

An emergent, multi-level model of the systems development process

Lynley Joy Hocking BA, BComp(Hons)

**Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy**

University of Tasmania (June 1998)

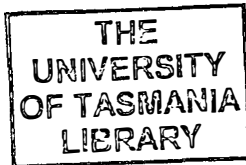
School of Information Systems.

This dissertation contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the Thesis, and to the best of the candidate's knowledge and belief no material previously published or written by another person except where due acknowledgment is made in the text of the Thesis.

This Thesis may be made available for loan and limited copying in accordance with the *Copyright Act 1968*

Lynley Hocking

June, 1998



Cent
Thesis
HOCKING
PLD
1998

Abstract

While systems development projects have become associated with increasingly sophisticated technologies and organisational changes, the models underlying the process have not kept pace. The Legislation Systems Project (LSP) is an example of a recent project where computerised technology was employed to improve the creation of and access to legislation. This project was substantially delayed due to unforeseen technical complexity and user acceptance was an ongoing concern. It will be demonstrated that the underlying model of process, which broadly reflects the dominant model of current best practice, is strongly linked with these problems.

This dissertation introduces and describes a multi-level process model which provides an in-depth exposé of the systems development process. This model reflects the closely intertwined nature of the content, context and process of change (Pettigrew 1985) and the interconnected relationship between the micro, macro and institutional levels of change.

As illustrated by descriptions of the LSP, the process can be viewed as the negotiation of meaning and interest at a micro level. This involves active participation, the creation and sustainment of coalitions of commitment and the application of authority and expertise. This analysis builds on the work of Walsham and others who have suggested the process of systems development should be viewed primarily as social interaction.

However, there are other levels of analysis which should be considered. Process implies change over time and Giddens' (1979) three levels of temporality provide a framework for a multi-level model of the systems development process.

At an institutional level, Giddens' structuration theory provides a useful framework for describing the manner in which change is deeply embedded in its organisational context and facilitates the creation of new social institutions. Both this and the micro-level processes can be described as emergent, both over time (chronologically) and in terms of social relationships (hierarchically).

This dissertation argues that the commonly used model of the process at a macro level, the systems development lifecycle, is inadequate as it ignores the emergent nature of the process at the micro and institutional level, the ongoing influence of the organisational context and the socially negotiated nature of change. However, there are existing alternative macro-level process models which do reflect both chronological and hierarchical emergence and this dissertation suggests some alternative ways of viewing system development initiatives such as the LSP.

Acknowledgments

Firstly, the participants in the case studies, especially those involved with the LSP, for contributing their time and allowing me to observe and review their activities.

Secondly, my supervisors, Dr Chris Keen and Dr Bill Ryan for their invaluable advice and support.

Thirdly, the members of my research group and wider research community for advice, comments on drafts, and discussions on pertinent issues.

Lastly, but most importantly, my family and friends, for just being there.

Table of Contents

ABSTRACT	3
ACKNOWLEDGMENTS	4
PART 1 INTRODUCTION AND BACKGROUND.....	1
1.1 RESEARCH QUESTION AND OBJECTIVES.....	2
1.2 RESEARCH CONTEXT	2
1.3 SUMMARY OF CONTRIBUTIONS	3
1.4 UNDERLYING AXIOMS AND RESEARCH PREMISES.....	5
1.5 RESEARCH METHODOLOGY	11
1.6 THE STRUCTURE OF THIS DISSERTATION.....	19
PART 2 CONTEXT – SYSTEMS DEVELOPMENT AS A NON-DISCRETE ENTITY	21
2.1 COMPUTERISATION, THE PRODUCTION OF LEGISLATION IN AUSTRALIA AND THE LSP.....	22
2.2 CONCEPTUALISING THE ORGANISATIONAL CONTEXT	39
2.3 THE ORGANISATIONAL CONTEXT - THE USERS	51
2.4 THE ORGANISATIONAL CONTEXT – THE SYSTEMS DEVELOPERS.....	72
2.5 REFLECTIONS: DEVELOPING A FRAMEWORK FOR EXAMINING SYSTEMS DEVELOPMENT AND ITS ENVIRONMENTAL CONTEXT – THE RELATIONSHIP BETWEEN GIDDENS’ STRUCTURATION THEORY AND PETTIGREW’S CONTEXTUALIST ANALYSIS	85
3. CONTENT – ESTABLISHING THE OUTCOMES OF THE CHANGE PROCESS.....	91
3.1 SYSTEMS DEVELOPMENT PLANNING AS A POLITICAL PROCESS	93
3.2 ACTIVE AND PASSIVE USER INVOLVEMENT	114
3.3 SYSTEMS DEVELOPMENT AS THE NEGOTIATION OF MEANING AND INTEREST	132
3.4 CREATING AND SUSTAINING COALITIONS OF COMMITMENT OF SUPPORT	155
3.5 THE APPLICATION OF AUTHORITY AND EXPERTISE	168
3.6 REFLECTIONS: THE LSP AND CHANGE IN THE OPC.....	187
4. PROCESS – IS PLANNING, DEVELOPMENT AND IMPLEMENTATION	207
4.1 CHANGE MANAGEMENT AND ITS RELATION TO SYSTEMS DEVELOPMENT	208
4.2 THE SYSTEMS DEVELOPMENT LIFECYCLE MODEL OF THE PROCESS	225
4.3 CHRONOLOGICAL AND HIERARCHICAL EMERGENCE	244
4.4 ALTERNATIVE MACRO MODELS OF PROCESS.....	276
4.5 SOCIAL INSTITUTION AND ACTION: A MULTI-LEVEL MODEL OF SYSTEMS DEVELOPMENT CONSISTENT WITH GIDDENS’ STRUCTURATION THEORY	295
5 REFLECTIONS AND IMPLICATIONS	312

5.1 SUMMARY OF CONCLUSIONS.....	314
5.2 SPECIFIC CONTRIBUTIONS.....	322
5.3 IMPLICATIONS FOR THEORY.....	323
5.4 IMPLICATIONS FOR PRACTICE	325
5. 5 FUTURE RESEARCH.....	329
5.6 CONCLUDING REFLECTIONS.....	329
APPENDIX 1 REVIEW OF COMMONLY AVAILABLE SYSTEMS DEVELOPMENT LITERATURE (PUBLISHED SINCE 1990)	332
APPENDIX 2 OPC PROCESSES AT THE BEGINNING AND END OF THE LSP.....	333
APPENDIX 3 BROAD CONCEPTUAL OVERVIEW OF THE ENACT SYSTEM	342
BIBLIOGRAPHY	344

Part 1

INTRODUCTION AND BACKGROUND

Near the beginning of a typical systems development project, an analyst focuses on the work processes of the client organisation and how they can be improved. If one were to conduct such an analysis of systems development, what processes would it describe and what improvements would be suggested? How would the existing normative descriptions of the processes be aligned with what the analyst perceived?

This dissertation can be likened to a systems analysis report of the process of systems development and implementation. It will suggest that the commonly-used descriptions of the process do not adequately reflect the actions of those involved, reflect some questionable underlying assumptions and, when used normatively, can detrimentally impact on the success of such projects. This dissertation will illustrate these points with reference to several core government information systems projects in Tasmania, the most significant being the Legislation Systems Project.

The Legislation Systems Project (LSP) is a major information systems development project in the Tasmanian State Service, designed to improve the production of and access to legislation. I tracked the project in depth over three and a half years through participant observation, interviews and reading associated documentation. When I started observing the LSP, the focus was the management of the associated organisational changes, which were anticipated to be far-reaching and challenging. These changes included work process changes, individual and organisational role changes and potentially cultural changes. How could or should those involved in the project manage these changes? How could organisational changes be integrated with the technical developments? IT enabled change has been a focal area of interest over the last decade but how does it unfold in practice? These were questions which intrigued me and challenged those involved in the LSP.

What was most challenging, but most intriguing for me, was that there were obviously no clear answers to these questions. Increasingly, it became clear that I, like many others, were asking the wrong questions. In essence, the real questions we were asking were, "How can *we* facilitate *their* acceptance of the changes *we* envisage?", while what we should have been asking was, "How can we facilitate changes that are appropriate in terms of the overall purpose, goals, environmental constraints and opportunities of the organisation and the requirements of those who affect and are affected by potential changes?". Pursuing this question led to a critical examination of how we view the process of developing and deploying information systems and technology.

1.1 Research question and objectives

The research question guiding this project has been:

What is the process by which core government information systems are planned, developed and implemented?

Associated research questions include:

- How is the process generally viewed and does this reflect the reality of how the process unfolds?
- How should we conceive the process of systems planning, development and implementation so it reflects what actually happens?
- How can we effectively manage this process?

The research objectives have thus been:

1. Identify important attributes of the process of implementing information systems projects in their organisational context of core public sector agencies;
2. Formulate a descriptive model which elucidates these attributes;
3. Analyse existing ways of conceiving the process; and
4. Suggest issues or areas for improving the management of this process.

1.2 Research context

In summary, this dissertation focuses on the organisational implications of information systems development¹, implementation and the effect of the organisational context on such activities. There is overwhelming evidence that most information systems implementation failures are non-technical and there is a recognised need for models of the change process which both capture the actual situation and are useful normatively (Hirschheim and Newman 1991; Buchanan and Boddy 1992; Clegg 1996). Human and organisational issues are recognised as the key issues for successful implementation of information technology in the future but most models of the process are simplistic or commonsense (Hirschheim and Klein 1989; Yates and Benjamin 1991). This project adds to a growing body of literature providing a richer picture of the process of developing and implementing information systems.

¹ Here systems development refers to the development of an information system within its organisational context, with software engineering being only a subset of this overall activity. Thus the acquisition of existing systems and their deployment into an organisation is also referred to here as systems development.

Research in this area is often classified according to whether it focuses on factors affecting the implementation of change or the processes by which such change occurs (Lucas 1981; Franz and Robey 1987; Myers 1995). Factor research produces conclusions based on contingency assumptions, such as "the more routine the technology employed, the more formal the organisational hierarchy will be" (Pfeffer and Lebleci 1977, Robbins and Barnwell 1994). This contingency approach can be criticised for adopting an objectivist ontology and a static view of organisations and for ignoring the process and ongoing nature of change (Barley 1990, Dawson 1994). In addition to not aligning with the underlying assumptions of this project, factor studies do not generally provide valuable insights. Significantly, they do not incorporate issues of time, negotiation, interpretation nor the often political nature of organisational change. Franz and Robey (1987) recommended the development of more effective process models, believing that further progress required "more complex, realistic models and the development of alternative perspectives for viewing implementation" (p 207). Almost a decade later, Myers (1995) also discussed the limitations of factor research and commented that the main finding of studies looking at the processes of organisational change was that the process mattered. Hence, there is still a need for good process models in this area.

The problem with most existing process models is that they ignore the close and intertwined relationship between the process, content and context of change (Pettigrew 1985). Systems development is essentially the implementation of organisational changes which include three primary elements:

- content (*what* is actually changing - for example, the implementation a new technical system);
- context (the *environment* in which the changes take place); and
- process (ie *how* change occurs ie the project strategy employed).

The interaction between these three elements is important, yet most studies tend to ignore the closely intertwined nature of these three aspects (Pettigrew 1995). The distinction between the context, content and process of change and a recognition of the close relationship between them forms a theoretical framework for looking at issues relating to the research question.

1.3 Summary of contributions

This project makes contributions on a number of levels. Particularly, it provides a detailed insight into several significant systems development projects and provides a critical evaluation of the process models underlying them and much of the normative literature in this area. The focus of these discussions is organisational change

associated with the implementation of information systems, an area of great interest to practitioners and researchers.

Three case studies are examined. The Legislation Systems Project is the most detailed and is based on a real-time longitudinal ethnographic study, a rarity in itself. A project to aid the production of and access to legislation, the LSP is a significant focus for study for several reasons, including the very nature of the project and its context of one of all governments' core functional areas. Such real-time longitudinal studies are a rarity due to difficulties obtaining access to such projects over a long period of time, but provide a deep insight into the processes of change as they occur.

The minor case studies track the development of two multi-department human resource management systems retrospectively. They are used to illustrate the concepts of hierarchical and chronological emergence before discussing them in relation to the LSP. Such large scale human resource systems are seen to be difficult systems to implement (Salmon and Proud 1993) and this project provides one solution for overcoming some of the problems by focusing on the process by which such projects are developed.

This change process can be viewed at several levels. At a micro level, the process can be viewed as one of social interaction. Giddens (1979) termed this the *durée* of temporality. He also identified two other forms of temporality: the *dasien* and the *longue durée* and this dissertation applies these concepts to the area of information systems development. Most of the literature on systems development refers to only one form of temporality, so the analysis of a systems development project according to these three levels of temporality is a significant theoretical contribution.

This dissertation identifies alternative process approaches which can be used to guide practice. At a micro level, this work supports that of Walsham (1993) and others who have argued the process is one of social interaction. Descriptions of the major LSP study illustrate that many of the actions of those involved could be described as the negotiation of meaning and interest with the associated application of authority derived from hierarchical relationships and/or expertise. At a more macro level, the process can be described as hierarchically and chronologically emergent. This contrasts with the dominant formal tools used by the systems developers, and several alternative macro process models are evaluated in terms of their ability to reflect these types of emergence. The dominant model of the systems development process is justifiably under threat and this dissertation provides some possible alternatives to it as well as grounding the criticisms of this systems development lifecycle model in observations of the three case studies.

This dissertation suggests that, in practice, there is a need to recognise the often emergent nature of change in organisations. This emergence occurs both chronologically and hierarchically and can be facilitated through iteration, involvement and incrementalism. These practices have implications for the common approaches to project management and systems development. Existing methods of project management do not cope well with emergent change issues as they tend to be based on the systems development lifecycle model, despite the existence of possibly viable alternatives. This suggests that alternative models should be employed if organisational change is an issue. Alternatively, it suggests that if we do not cope with emergent change adequately within specific projects, the scope of IS development projects must be much smaller than they tend to be now. We should think of major change initiatives as involving cycles of projects, rather than a project cycle, as is often the case now. In other words, there is a real need to carefully consider the scope of projects and/or ways of approaching information systems development projects. The contributions of this dissertation are discussed further in the conclusions.

1.4 Underlying axioms and research premises

Axioms are the meta-theoretical assumptions about the nature of research and reality which guide social actors, be they researchers or systems developers. Everyone makes assumptions about the way the world works and their relationship to it. These assumptions guide and influence our research approaches and outcomes, and so should be explicitly examined (Hirschheim 1985; Franz and Robey 1987; Keen 1991; Orlikowski and Baroudi 1991).

Assumptions concerning the nature of reality (ontology) and our relationship to it (epistemology) greatly impact on the process and outcomes of a research project. Given the multitude and conflicting nature of research conclusions in this area, such assumptions should be examined and closely tied to the rest of the project. The discussion here justifies a broadly interpretivist epistemological stance employing Giddens' structuration theory at an ontological level.

1.4.1 Ontology

Ontological questions are concerned with the nature of reality and two broad stances are commonly identified (Burrell and Morgan 1979). In adopting an objectivist stance, one would view the social world as being made up of tangible objects which are just as concrete as the physical world. Adopting this stance often leads to a focus on social structures as entities in their own right and an examination of their impact on individuals' actions. Subjectivists (nominalists) believe that social structures are

only created through people's perceptions and actions. Those who adopt a subjectivist ontological stance, such as symbolic interactionists, tend to focus on how human actors create social structures (Walsham 1993).

Burrell and Morgan (1979) argued these two ontological stances are mutually exclusive. This idea of strict ontological incommensurability has stimulated a great deal of debate, with some arguing that a strict division between ontological paradigms does not make sense or is harmful as it freezes debate (see for example Gioia and Pitre 1990; Jackson and Carter 1991; Ackroyd 1992; Aldrich 1992; Willmott 1993; Weaver and Gioia 1994; DeCock 1995). Giddens' structuration theory effectively resolves this debate by illustrating that the relationship between the two stances is a dualism, rather than a duality (Poole and Van de Ven 1989; Weaver and Gioia 1994).

Structuration theory illustrates that change involves a process of "structuration" as new structures are created and reconstituted through peoples actions and perceptions (Giddens 1979; 1989). It is an extremely rich and potentially useful process theory (Barley and Tolbert 1997) and is used to provide a broad theoretical framework for this research project and inform the description of change at an institutional level. It effectively illustrates how the changes involved with information systems (IS) implementation are deeply embedded in the status quo and how actions involved with the implementation of IS can create new social structures through the creation or facilitation of new roles and tasks. That is, it effectively illustrates the nature of change in organisations, a theme at the heart of this dissertation.

Giddens defined structuration as "the conditions governing the continuity or transformation of structures, and therefore the reproduction of [social] systems" (p 345). A key concept of the theory is that social structures are not viewed as a framework as it commonly is, but as "rules and resources" which are implemented during social interaction. Hence, structures are not seen as something independent of human actions, but phenomena created via their actions. At the same time, such structures have an enabling and constraining influence on these actions. Actions are defined as the continuous flow of interventions in the world by social actors.

At a more detailed level, structuration theory includes three modalities linking the structure of organisations with human interactions. As shown in Figure 1.1, the structure of organisations consists of modalities of signification, domination and legitimation while human interaction consists of modalities of communication of meaning, power and moral sanction. These elements of structure and interaction are linked respectively by interpretive schemes, resources and norms. Interpretive schemes include the shared knowledge which people use to interpret behaviour and events and to communicate meaning. Resources are the means through which

intentions are realised, goals accomplished and power is exercised. Norms are the rules governing behaviour by defining appropriate conduct which sustain and articulate the structures of legitimation (Orlikowski and Robey 1991).

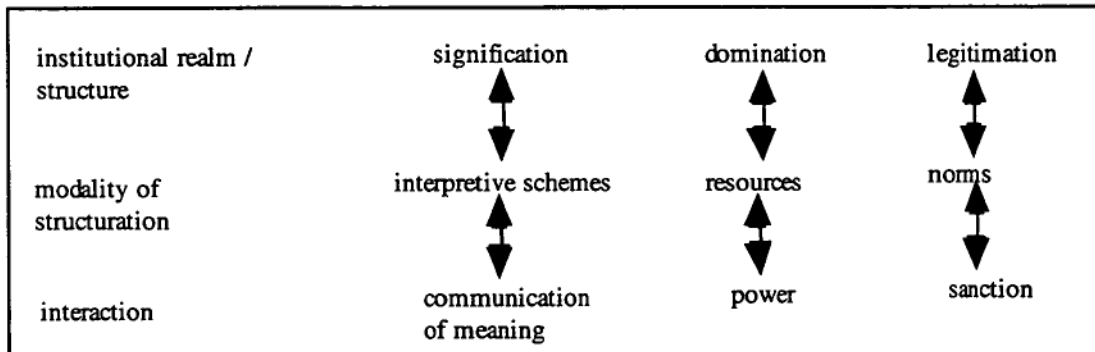


Figure 1.1: The modalities of structuration theory (source: Orlikowski and Robey 1991; Walsham 1992)

Structuration theory has been used by many looking at organisational change and technological change in particular (eg Scott Poole and DeSanctis 1990; Walsham and Han 1991; Jones and Nundakumar 1993; Knudsen 1993; Han 1994; Shanks, Hodgson et al. 1996; Dillard and Yuthas 1997). For example, Orlikowski and Robey (1991) focused on its ability to incorporate technology and technological changes in organisations. Specifically, they used the above framework to look at information systems development and information systems use. They found the framework was useful because it:

- included the intended and unintended consequences of human actions;
- looked at how human actions and organisational structures shaped the technology; and
- focused on how technology shaped them and included historical and contextual factors.

Walsham and Han (1994) illustrated the theory could explain the processes of strategy formation in a central government agency by illuminating the close interconnection between social action and social structures through the consideration of meaning, power relations and values. Dillard and Yuthas (1997) interpreted the decision making process in a large accounting firm using a number of different approaches and argued that structuration theory gave a much better insight into these processes and their outcomes than other dominant approaches. Evaluating the literature in the area and reflecting on its use with a number of case studies, Shanks et al (1996) concluded structuration theory was of great value in analysing and interpreting data in empirical research.

However, structuration theory is very abstract and is difficult to apply directly to case studies (Giddens 1989; Gregson 1989). Gregson (1989) defined it as a "second order" theory which helps explain social phenomena in general terms but is difficult to transfer to empirical studies. Giddens (1989) preferred the term, "sensitising device" to describe the role of structuration theory in relation to empirical observations, and that is the way it is primarily employed here. This issue is pursued further in Chapter 4.5.

Structuration theory is of great relevance to this research project, focusing as it does on "recurrent social practices and their transformations" (Giddens 1991: p 203). While recognising the criticisms and debates surrounding the theory (eg Bryant 1991; Gregson 1989), its growing acceptance and conceptual depth makes it a good "sensitising device" and ontological framework. It is discussed further at the end of Parts 2 and 4.

1.4.2 Epistemology

Epistemological questions are concerned with how we can perceive or research reality. Epistemology is closely linked with ontology but, while structuration theory provides an appropriate ontological stance, Giddens does not focus specifically on epistemological issues. Giddens did suggest that neither a positivist nor interpretivist epistemology is entirely suitable if one adopts structuration theory, but has not fully articulated an alternative (Bryant and Jary 1991). Hence, while employing structuration theory here, I have adopted an interpretivist epistemological stance. Here this stance is justified.

Orlikowski and Baroudi (1991) suggested studies could reflect one of the following epistemological stances:

- positivist;
- interpretivist; or
- critical.

They suggested it was important for researchers to examine and justify their epistemological stance, a suggestion which is pursued here.

Most research in this area is positivist and assumes there are fixed relationships within a phenomena which can be investigated using structured instruments, formal propositions and quantifiable measures. Positivist studies aim to make inferences or generalisations to other situations by looking at regularities in or between phenomena. Positivism is not appropriate for studying the implementation of IS in organisations because:

- it assumes a fixed/stable reality while reality is emergent and socially negotiated (Truex 1995);
- it does not take the standpoint of the researcher into consideration;
- the search for universal laws tends to make researchers ignore historical or contextual conditions (Orlikowski and Baroudi 1991); and
- it ignores the fact that people are active creators of their physical and social surroundings (Orlikowski and Baroudi 1991; Nandakumar 1993; Myers 1995).

Thus positivism is not appropriate for this study.

Critical theory has become popular among some IS researchers recently (eg. Hirschheim and Klein 1989; Hirschheim and Newman 1991; Jackson 1992; Klein and Hirschheim 1987; Klein and Hirschheim 1993; Ngwenyama 1991). Klein (1992) described three areas where critical theory was being used in IS research:

- criticisms of the instrumental rationality underlying IS and the dominant management ideology (eg Klein and Hirschheim 1987);
- criticisms of the dominant research paradigm used in IS research (eg Hirschheim and Klein 1988, Winograd and Flores 1986, Ngwenyama 1987);
- classifications and criticisms of existing technology-driven models and an explanation of alternatives (eg Lyytinen and Klein 1985; Klein and Hirschheim 1993).

In a seemingly unrelated paper, Jackson (1992) listed the five "pillars" of critical theory and illustrated how they had been addressed in the IS literature by authors such as Hirschheim, Lyytinen, Checkland, Walsham and others. Critical theorists critique the *status quo* by trying to expose what they see as deep-seated structural contradictions and attempt to transform the situation so that people are no longer alienated from their potentialities due to economic, political and cultural systems (Orlikowski and Baroudi 1991).

While this study does reflect some critical elements, a purely critical epistemological stance is not appropriate for this research topic for several reasons:

- Pure critical theory ignores the fact that people create and maintain their social world and treats the social world as separable from the people who create it (van den Berg 1980); and
- Critical theorists assume that the concepts of freedom (emancipation) and reason are inextricably linked. As Lucas (1985) argued, if there is such a thing as *a priori* reasoning then a person cannot be totally free to choose. Likewise, if a person is totally free to choose, they must also be free to choose irrational (or non-

emancipatory) options. Additionally, the concept of emancipation can be equated with freedom, yet freedom can be interpreted as the freedom *to* do something or the freedom *from* constraints (Berlin 1969).

- Pure critical theorists maintain that there is a higher truth (emancipation, freedom from alienation) like positivism, but reduce empirical studies to value judgements. Unlike interpretivists, though, they maintain that these "higher truths" are able to be verified non-empirically, yet this process of verification is not discussed (van den Berg 1980) .

Van den Berg (1980) was scathing of critical theory generally. He described it as "a way of rendering Marxism indeterminate and hence unfalsifiable". He saw little relationship between theory and practice in critical theory and believed there had been adequate discussion of issues such as the creation of a just society and the alleviation of injustices. For him critical theorists added nothing to these debates and have only contributed to them from "the comfortable heights of philosophical abstraction" (p 476). While not so critical of the approach, Orlikowski and Baroudi (1991) also stated that critical theorists tend to not be critical enough of their own concepts and theoretical models. Hence, this study does not adopt a purely critical stance, though it does include some critical elements.

Based on an subjectivist ontological stance, interpretivists assume that, as people create and sustain meanings and concepts through their interactions, the role of the researcher in the field cannot be value-free. Interpretivist studies do not aim for statistical generalisability. As Walsham and Waema (1994) pointed out, the validity of this approach depends not on statistical extrapolation, but on the plausibility and reasoning of the description and explanation of the cases. Interpretive studies generally focus on more complex phenomena than positivist studies and aim to locate their findings in the context and, recognising that no two social situations are ever exactly the same, it is pointless to try and aim for statistical generalisability.

Extrapolation from one or more individual cases is not achieved statistically, but in a theoretical sense, as plausible concepts derived are applied to other scenarios and observations are abstracted to theoretical concepts. The approach has gained acceptance for looking at the role of technology in organisations (Walsham 1995) and is employed here.

Nevertheless, there are some shortcomings with adopting an interpretivist approach. Interpretivists often do not explain the unintended consequences of action and tend to ignore structural conflicts and contradictions (Giddens 1979, Orlikowski and Baroudi 1991; Nandakumar 1993). However, "any way of seeing is also a way of not seeing" and all epistemological approaches have some shortcomings. These criticisms are

however less fundamental than the ones aimed at the positivist approach, and can be at least partly counteracted through the choice of an appropriate research methodology.

1.5 Research methodology

The methodology employed can impact on the contributions and conclusions of a research project and so should be explained. Broadly, this project started as what Millar and Friesen (1982) refer to as a real-time, broadly focused, non-quantitative longitudinal study of a single organisation with retrospective non-quantitative observations of two minor case studies focusing on particular issues. In the primary study, data was collected via participant and non-participant observation, interviews and the examination of associated documentation. I used interviews and document analysis to retrospectively form an historical account of the secondary case studies. As this project utilises structuration theory as an ontological sensitising device, it should employ longitudinal, historical and contextual research methods (Barley 1986, Orlikowski 1988). Giddens argued that structuration theory helps mitigate much of the conflict between quantitative and qualitative research methods (Orlikowski 1988) but qualitative methods are primarily employed here due to the nature of the research questions.

The choice of research methodology largely depends on the research question and the underlying axioms of the study (Keen 1991). Focusing on the process by which technologically-induced change unfolds in organisations, this project does not lend itself to quantitative research methods as it requires a more holistic approach than such methods would allow. Organisational changes are highly dependant on their context and, by focusing only on specific elements, quantitative research methods can easily lead to oversights. Quantitative methods are generally not compatible with an interpretivist epistemological stance as they:

- postulate the existence of fixed causal relationships between variables;
- view reality as stable; and
- perceive social structures as existing independently of the people involved.

These are characteristics of a positivist epistemology. Such studies often focus on unidirectional causal relationships, have produced contradictory results and have been based on the doubtful assumption of mechanical causality (Orlikowski 1988; Brinckman 1991; Kraemer 1991; Kling and Dunlop 1993). In-depth, longitudinal non-quantitative studies are recommended for research into organisational change (Pettigrew 1985; Dawson 1994).

The qualitative approach used here is analysis using real-time ethnographic research methods using both participant and non-participant observation, retrospective analysis via interviews and an examination of project documentation (Yin 1989). The use of a single primary study relates to an interpretivist stance and the minor case studies are not included to suggest statistical generalisability but to illustrate general propositions in a less abstract manner. The major case study, the LSP, was investigated via ethnographic methods to create a “thick” description of events and issues (Atkinson and Hammersley 1997). I have not tried to completely empathise with the subjects in the case study as the many differing perspectives and world views would have made this impossible and limiting. However, I have tried to understand what Geertz (1988 in Harvey and Myers 1995) termed the “webs of significance”. In other words the thick description of the LSP aims to reflect the opinions and worldviews of those directly involved with the LSP. Participants’ responses to early drafts of sections of this dissertation suggest that I have been able to do so.

I observed the LSP for three years and a half years, with concentrated periods of involvement in the first six months and periodically throughout the project. I attended meetings, shadowed participants, conducted informal and formal interviews, observed activities, examined documentation and was sometimes actively involved. Here specific issues are described in order to address concerns regarding rigour and validity. Specific issues which should be considered include:

- the use of an in-depth longitudinal ethnographic study;
- real-time versus retrospective analysis;
- demonstrable rigour;
- the choice of research site;
- my involvement in the field;
- the process of data analysis;
- the use of theory in relation to observations from the field; and
- the omission of data observed from descriptions.

These are examined in more depth below.

1.5.1 An in-depth longitudinal ethnographic study

Real-time longitudinal studies are the best way to look at change in its organisational context (Franz and Robey 1987) but are uncommon in information systems research. This is primarily because they can be costly, involve a great deal of extended effort and are difficult to carry out as obtaining access to a research site over an extended period can be difficult (Vitalari 1985). However, they also have significant

advantages over single data collection studies as they permit a better exploration time-dependant phenomena such as change (Vitalari 1985). Although not common, there are a number of well-known longitudinal studies of single cases. For example, Orlikowski (1988) investigated the diffusion of CASE tools in an accounting/consulting firm focusing on issues of power. Such studies are widely referred to, having provided significant insights into this area of study.

1.5.2 Real-time versus retrospective analysis

While the secondary case studies retrospectively reconstruct events from interviews and documentation, the in-depth primary ethnographic study described here tracked changes as they occurred over time. The advantage of the former method is that it is quicker and does not involve wading through vast amounts of detail. The disadvantage is that events reconstructed retrospectively tend to ignore the ambiguity and untidiness in real world situations as they unfold. In recounting events, people tend to retrospectively tidy them into a logical progressive account. Simply, the secondary case studies do not provide anywhere near the depth as the primary ethnographic study. Further illustrating the advantages and disadvantages of each approach, the secondary case studies also took much less time and effort, with the primary ethnographic study involving about 120 days of observations alone². There were plans for several other secondary case studies, but these have proved unnecessary as continued access to the primary case study was achieved.

1.5.3 Demonstrable rigour

As Applegate and King (1997) point out, the intellectual foundations of case study research are robust. Demonstrable rigour is achieved by making data gathering and analysis techniques explicit and transparent, as is done below. In addition to this, the validity of the descriptions and discussions of this research project are enhanced by:

- aiming to find multiple sources of evidence for each observation (Yetton, Johnston et al. 1994);
- actively looking for negative evidence (Dey 1993);
- feeding back observations and interpretations to those in the field (Pettigrew 1995; Rusli and Marshall 1995);
- maintaining a log of events and interpretations to aid reflection; and
- ongoing analysis through discussions with research colleagues and supervisors.

² From the limited information available, this seems similar to the efforts of (Orlikowski 1988; Barley 1990; Barley 1995).

1.5.4 The choice of research site

The LSP was chosen for two primary reasons. Firstly, it involved a large degree of change from the perspective of the users. This was the focus of my research and the questions being pursued in this research project were also being asked by those involved in the LSP. Essentially, the LSP provided a slightly caricatured or archetypical example of the themes being pursued, and my access to the project was facilitated by a common interest in these themes. Secondly, this case study is very interesting to many different groups due to the nature of the system, its interaction with its context (that is, political and legal systems) and the technology involved and it has been valuable to see the system unfold. The LSP is an important information systems development project, providing insights into significant conceptual issues.

1.5.5 My involvement in the field

It is important to reflect on the role of the researcher and how this may have affected the phenomena under investigation (Prasad 1993; Rusli and Marshall 1995). As in any research project, it was impossible not to interact and be part of the observed phenomena even when trying to be a non-participant observer. Giddens (1989) termed the way in which the actions and perceptions of the researcher become part of the social interaction being observed the "double hermeneutic cycle".

On one level, the double hermeneutic cycle describes how "lay members of society routinely reincorporate social science concepts into the world" (Held and Thompson 1989: p 251). When people in the situation under investigation, or similar situations, start using the concepts developed by researchers, it illustrates the utility and plausibility of the concepts developed by social researchers. Hence, towards the end of this longitudinal project, the concepts being developed here were presented to those involved in the study for comment. As well as being appreciated by those in the field, this was a good validity test.

The double hermeneutic cycle can also occur within social research projects, so that the actions and perceptions of the research directly influence their observations and results (Dey 1993). Interpretivists recognise this is unavoidable but that it must be at least partly addressed by actively analysing the role of the researcher as part of the research context. Issues which must be considered include:

- the role of the researcher in the social context;
- attributes of the researcher which influence their access to information;
- access to individuals and information due to that role; and
- assumptions and world views of the researcher.

In this project, my role changed over the time of my involvement in the field and occasionally I played several roles at the same time. I initially took an active role, but my involvement gradually became more passive until the last eight months of my involvement when I held a position in a closely related unit. During this time, I was not directly involved in the LSP, but was actively involved in other capacities as described further in Part 5. As well as working alongside some of the systems developers to create in-house project management guidelines, I discussed my observations with those involved in the project to check if they aligned with their interpretation of events. This proved a useful exercise, not only as it helped check the validity of the descriptions, but also because people's reactions sometimes provided further insights into how particular groups or individuals interpreted events.

Nundakumar and Jones (1997) highly recommended this participant observation research approach but noted it was only employed by 1% of papers published in widely disseminated papers between 1993 and 1996. They suggested this might be because the social dynamics in the field can be difficult to manage. Arising issues include:

- confidentiality;
- problems of obtaining access to all or some relevant sites;
- the influence of the researcher on the phenomena under investigation;
- changes in the organisational context may require a change in research focus;
- a large amount of time is required for field research; and
- the researcher has to juggle the dual role of trying to appreciate the day-to-day experiences of the social actors but also continually question hidden assumptions.

These were also issues I faced in the field and some of the methods for addressing them have been or are discussed below.

Even as a non-participant observer, it was impossible to remain neutral in a social situation being observed and not influence it. For example, people would seek my opinion about the project or aspects of it and those involved in the development project seemed more aware of the issues I was investigating when I was around. For instance, one time one of the systems developers turned to me and said, "See – we are letting them (the users) make the decisions here". These influences may have impacted on my access to information, but it is difficult to see how they could have greatly influenced my broad observations and conclusions.

1.5.6 The omission of data observed from descriptions

Involvement in the field did not just involve the collection of data, but its initial interpretation as well. While the description of the LSP here aims to provide a detailed insight into the project, the overwhelming wealth of available information had to be distilled to provide meaning and relevance. Reasons for omitting data included:

- *a lack of physical accessibility.*

For example, when the main technical components of the project were being developed in Melbourne, an aeroplane flight away, my access to the project's development was greatly limited to perusing correspondence and interviews with participants until acceptance testing. Additionally, when the project faced long delays, it became more difficult to obtain access. This was partly because few project activities at this time occurred in Hobart, but also due to increasing concerns the project would not be a success. As the project progressed towards what looked like a successful implementation, it became easier again for me to obtain access. This access was also facilitated by the fact I was now working in the same offices as the systems developers for half the working week;

- *the wealth of detail.*

"Thick" descriptions result from attention to details, but it would be easy to become lost in a maze of detail and lose sight of the overview broader theories can provide. The fact that hours were spent discussing new standardised formats for amendment legislation may be pertinent but most of the details of such meetings are not. Hence my own thick descriptions are still summaries;

- *relevance.*

Issues were discussed which were not pertinent to this study. These include issues not relevant to the LSP and issues which were part of the LSP but were not relevant to the theoretical area under investigation, such as technical details of the project, details concerning data conversion activities and so forth;

- *confidentiality/ potential to cause harm to individuals.*

Most of the time individuals are referred to by their job descriptions to preserve anonymity. The personal attributes people bring to a job can influence their ability to do it but this is largely ignored here. Information or analysis which could potentially harm individuals is not described here, but this should not greatly influence the theoretical development of the project, as such issues were still considered only not reported. I was also asked to sign a confidentiality agreement

to ensure I would not discuss potentially sensitive information, such as draft legislation;

- *my role in the project/perspective on the project.*

"What one sees depends on where one stands" and my observations are bounded by my own assumptions and viewpoint. This issue cannot be sidestepped or avoided, but is addressed by the examination of the assumptions underlying this project to a degree. Other assumptions which might influence my perspective are my background in information technology and my alignment with the interests of the systems developers at times. I have tried to address this issue by trying to make my standpoint explicit.

1.5.7 The process of data analysis

Data analysis involved two primary steps. The first was the distillation of the wealth of information derived from the case studies into a rich description of incidents and issues of the project as it progressed via a process of summarisation. The second step involved the consideration of these incidents and issues against existing and emerging themes. These two steps have been iterative, and this is reflected in the structure of the thesis.

Field notes provided an abundance of detail and in order to gain some initial insight into it, it was necessary to try and summarise them. Thick descriptions are an important part of qualitative observations, forming the foundation for further theory building (Dey 1993; Walsham 1995). These thick descriptions display the subtleties and some of the multiple perspectives which make up "reality". Many of the recent developments in the social study of technology have produced thick descriptions which have provided a wealth of detailed information, but this needs to be structured if it is to be of use (Bijker, Hughes et al. 1989).

The structure of the thick description was not closely aligned to existing theoretical frameworks and emerged over time. I initially took a grounded approach, with observations from the field categorised initially according to incidents and issues which the systems developers and/or users had to contend with as the project progressed. Examples of such incidents and issues include:

- the fact that the impetus for the system came from outside the user organisation and the manner in which user commitment to the concept of the system was obtained;
- the complicated nature of the drafting process and the actions the systems developers took to gain an adequate understanding of this process;
- the lack of user experience with computers and steps taken to reduce or avoid this issue, such as the use of prototypes and training on the technology; and

- the development and promotion of a user developed prototype and the reactions of the systems developers.

These incidents and themes were used to structure the thick description that is woven in with the theoretical discussions in this dissertation.

Over time, some of these incidents and issues seemed to naturally group together around certain themes. Miles and Huberman (1994) refer to this activity as noting patterns and themes and clustering them. For instance, the latter three example incidents mentioned above all seemed to concern some kind of learning, a concept I later refined to the negotiation of meaning.

These themes partially emerged from reflection on the incidents and issues at the first level of analysis, but were also influenced by concepts in the existing literature. One of the criticisms of using existing theories is that it is very easy to only view field observations through the lens of the existing theory. However, this was avoided by not being committed to a particular theory or set of theories until relatively late in the project. The wealth of existing theories focusing on technological change in organisations provided a range of lenses through which to view the project. Some of these theories were discounted because they did not comply with the underlying axioms of this project, assuming, for example, a deterministic relationship between technological change and changes in organisations or an objectivist view of organisations. Some theories seemed to fit the observations from the field better than others, as will become apparent below.

Over time links between observations from the field and certain theories were strengthened. For example, the concept of learning gradually evolved into the learning and negotiation and then into the concept of the negotiation of meaning and interest. This iterative consideration of theory and observation is reflected in the structure of this thesis, with discussion of theoretical concepts and existing literature interspersed with observations of the LSP. An iterative approach to theory development is thus highly recommended (Jorgensen 1989; Layder 1993; Miles and Huberman 1994; Walsham 1995) and has been adopted for this research project.

In the latter stages of the research project, ongoing data from the field and reviews of my notes and other documentation were used to strengthen or discard emerging concepts. During this stage, concepts become better articulated and were sometimes adapted to ongoing observations or further information from other sources. Miles and Huberman (1994) referred to this as building a logical chain of evidence and making conceptual/theoretical coherence. The final stages of selective coding were combined with writing this dissertation and discussing it with research participants and colleagues.

1.6 The structure of this dissertation

This dissertation broadly follows what Yin (1989) describes as a "theory building" structure, with the literature and initial findings from the field research being discussed alongside the descriptions of the theories employed. As well as reflecting the iterative development of research themes through the examination of relevant literature and the case studies, the broad structure of this dissertation uses Pettigrew's (1985) identification of the three fundamental elements of organisational change as a framework for discussions. That is, Parts 2, 3 and 4 focus on issues of context, content and process respectively. Part 2 examines the issues of organisational context and the fact that systems development projects cannot be viewed as discrete from their organisational contexts. In Part 3 the focus is on how the content of such change projects is defined, illustrating the social interactions which make up the process of development at a micro-level. Moving to a more macro-level, Part 4 concentrates on the manner in which the process of systems development is conceived and will look at dominant and alternative models of process. It also touches on the process of change at an institutional level. Theoretical discussions are interspersed with ongoing descriptions and analysis of the LSP and the minor case studies to illustrate points made.

This iterative and emergent structure has been employed for several reasons. Firstly, it echoes the emergent manner in which this research project has unfolded, and the way theory has been iteratively used in relation to observations from the field. Secondly, and most importantly, it reflects the process by which information systems initiatives such as the LSP or the two HRMS case studies unfold in practice. Finally, this structure allows a consideration of relevant themes and their immediate illustration through descriptions of incidents from the case studies.

Part 2

CONTEXT

– systems development as a non-discrete entity

This thesis tells the story of a major systems development project and draws implications from it. This story has to be placed in context and Part 2 sets the scene, so to speak, for the rest of the dissertation. Firstly, it introduces the Legislation Systems Project and places it in its broad context of computerisation and the production of legislation in Australia in Chapter 2.1 Chapter 2.2 goes on to examine the reasons for looking at the context of systems development and illustrates that the perspective one has on the context can greatly influence how one views the process. It concludes that there must be an adequate focus on the social as well as the technical and process aspects of the organisational context of systems development. Chapter 2.3 then identifies and articulates important relevant contextual issues which should be considered in relation to particular case studies and systems development projects. These aspects are explored in relation to the primary clients of the Legislation System Project, the Office of Parliamentary Counsel, and implications for the systems developers are suggested.

At the same time, it is also important to reflect on the context of the systems developers and the differing frames of reference which influence their interaction with their clients. The organisational context and differing frames of reference of the LSP systems developers, as compared with the drafting office are the focus of Chapter 2.4, and, again, implications for the process of systems development are drawn.

Part 2 concludes by illustrating the integral and ongoing role that the organisational context of systems development plays in the process of developing and implementing information systems through projects. The final section of Part 2 also introduces Giddens' concepts of *durée*, *dasien* and *longue durée*, which are used as a basis for a multi-level model of the systems development process which effectively copes with the important and socially constructed nature of the context.

2.1 Computerisation, the production of legislation in Australia and the LSP

The major empirical focus of this dissertation, the LSP, is a significant systems development initiative, both in terms of its practical application and its relevance to systems development theory. Here the LSP is introduced and the impetus for it explained. An examination of legislation production in Australia and the use of computers to create and access legislation reveals that trends promoting the LSP are similar to other jurisdictions. However, the initiative is unique in that the resulting system aids both the creation of and access to legislation while in other jurisdictions they are considered separately. This introduction to the LSP places the project in context and helps "set the scene" for the process of developing the resulting EnAct system.

Access to consolidated legislation has been a problem in most government jurisdictions based on the Westminster model. Determining the state of statute law can be a time consuming and tedious task. Firstly one must obtain a copy of the original, or principle act, plus a copy of all the amendments made to it since it was passed. Then all the amendments must be manually noted on the principle act. In other words, one must manually consolidate the legislation in order to gain access to its content. Recently, commercial organisations have produced consolidated versions of legislation on CD-ROM, but these are consolidated on a periodic basis and so have to be constantly updated. Given that all citizens are expected to have a reasonable understanding of the law, this lack of access to consolidated legislation has been a significant issue for all Australian jurisdictions and other governments based on the Westminster model³.

The Tasmanian Government has recently resolved this issue by implementing the EnAct system and providing public access to the associated document database via the internet. This dissertation describes and analyses the process by which this

³ In other countries, such as the United States and Norway, Acts are replaced by an entirely new one if they are to be amended. This alternative way of changing existing legislation avoids the complication of having to consolidate legislation, but is said to produce delays in introducing amendments into parliament as the whole Act, rather than the amendments, is presented for debate. In other words there would be an increased tendency to rework existing provisions which were otherwise satisfactory, so impinging on the stability of the statute book.

There are obviously many aspects to this issue, but they will not be addressed here as they are outside the scope of this project. For the Tasmanian drafters and those involved with the LSP, the existing method of introducing Amendment Acts to change existing legislation was an objective, unquestionable fact. They believed that to introduce the alternative method would require substantial changes in parliamentary practice and this was outside the brief of the project.

system was planned, developed and implemented through the Legislation Systems Project (LSP).

In this chapter, the initiative is introduced and placed in context. Reflecting on the production of legislation in jurisdictions such as Tasmania and the use of computers in relation to statute law reveals that the LSP is a unique systems development initiative in that it covers both the creation of and access to statute law while these areas are considered separately in other jurisdictions. This examination also reveals the impetus for the EnAct system are similar to issues faced in other jurisdictions, so suggesting this initiative will have wider relevance.

The production of legislation is one of the core activities of government, legislation being the "framework in which governments achieve their purposes" (Crabbe 1993 quoted in Lim 1993: p 3). Considering the importance of legislative drafting in the workings of government and law courts, there is remarkably little written about the role of drafters, the processes of drafting and the influence of computers. Voermans and Verharen (1993) investigated the possibility of using knowledge-based systems as aids for drafting legislation but did not look at the actual processes of producing legislation and the effect of the proposed technology on it. Chartrand and Ketcham (1994) looked at the opportunities for using information resources and advanced technologies in the US congress but focused on the application of computerised tools to aid the consideration of legislation and information conveyance rather than the possible consequences of using the technology. Snellen and Schokker (1992) discussed the interaction between the creation of legislation and new information systems and the application of the law enabled by new information systems, but also did not focus on the effect of information technology (IT) on the production of legislation.

This dissertation builds on their work by describing the Legislation System Project. Although there are other potential applications of computers in relation to statute law, they have generally only been employed in relation to aiding access to law through information retrieval systems or as administrative support tools for the production of legislation. The uniqueness of the LSP largely derives from the scope of the project, which covers both these areas in an integrated manner. Although the LSP is a unique project, the contextual issues promoting the project are typical of similar jurisdictions. This chapter places the LSP in context and illustrates the relevance of this project to other jurisdictions.

2.1.1 The legislation production process in Tasmania and the LSP

Tasmania is the smallest state of Australia, with a population of approximately half a million. The Australian state governments broadly have the same functions as the Canadian provinces and operate according to the general principles of the Westminster parliamentary system. In Tasmania there are two houses of parliament:

- the House of Assembly, where the governing and opposition parties sit and the executive government is formally positioned; and
- a house of review, called the Legislative Council.

The production of legislation is a highly complex and iterative process in Tasmania and is similar to other Australian jurisdictions with two houses of parliament. Although this study is limited to an isolated example, it has implications for the legislation production processes of any government and so should be of interest to others involved in information systems development in the public sector or the production of legislation anywhere.

Here the formal aims and objectives of the LSP are described as a background to the rest of the project. These are derived from the project's business case document, the development of which is discussed in Chapter 3.1.

a) The production of legislation in Tasmania

Legislation includes a range of instruments. Acts of Parliament are the means by which government policy is most directly implemented (Lim 1993). Other instruments include subordinate legislation such as regulations, by-laws, proclamations and orders. These instruments do not generally pass through parliament but are administered by the department responsible. Here, the focus of discussions is on Acts of Parliament, though many of the same issues apply when considering subordinate legislation and the LSP covered both.

For the purposes of later discussion, it is worth clarifying some of the terms used. *Acts* have been passed by both houses of parliament and have received Royal Assent and so are considered a formal part of the state's law. *Bills* are acts being debated by the houses of parliament, and *drafts* are proposed legislation before it reaches the House of Assembly.

The legislative process involves two main sub-processes: drafting and enactment (Mason 1988). Drafting is the process of articulating policies in a written legal form. Enactment is the process of authenticating and approving documents so they can pass into law.

For Acts of Parliament, processes of enactment are basic parliamentary processes, where new legislation is presented, debated and passed. Most people are aware of the basic workings of parliament, at least in their own jurisdiction. In a democracy based on the Westminster model, new legislation (called a Principal Bill) or amendments to existing legislation (termed Amending Bills) are presented to one house of Parliament by an elected member of that house. The Bill is then debated and perhaps amended before being either accepted or rejected. Most of the time such Bills are introduced to the lower house before being reviewed by an upper house and in jurisdictions based on the same model but with only one house, there is no Parliamentary process of review. In the two house, bicameral system, both houses must agree to the contents of the legislation before it becomes an Act and can be enforced. Most legislation must be signed by the formal head of state (the governor or governor general) before it becomes part of the formal body of statute law. This is the model followed by all the states of Australia and the federal Commonwealth government, the federal and provincial governments of Canada and other governments based on the Westminster model. The enactment of policy into legislation is concerned primarily with political processes in and around parliament. These processes are largely outside the scope of this project and the LSP, but interact strongly with them.

Both this project and the LSP are primarily concerned with the background drafting processes and access to legislation once it passes into law. In many respects, these areas can be considered the administrative functions associated with the broader enactment processes.

As in other states of Australia, legislation is written by lawyers who specialise in the drafting of legislation, called Parliamentary Counsel or, less formally, drafters. Legislation is generally initiated by members of the governing party, usually through Cabinet and is further developed through consultation between the agency responsible for the area of policy covered by the legislation and the Office of Parliamentary Counsel (OPC) before it is reviewed by the Legislative Review Committee, printed by the Government Printing Office and debated in Parliament. The division between policy creation and drafting is not always clear but generally the instructing department is responsible for the policy articulated in a bill while drafters are responsible for its legal effectiveness (Mason 1988). Amendments made on the floor of either house of Parliament also must be included in the legislation.

McDonald (1993) provides a more detailed insight into this process, which is illustrated here diagrammatically. Figure 2.1.1 illustrates the process of producing such legislation through different governmental bodies before the LSP. The process was only slightly different for private members bills. Such diagrams outlining the flow of legislation from its initiation until receiving Royal Assent were produced by

the LSP team and are believed to be part of the first thorough review of the process since it was initiated in Tasmania in 1854.

b) Impetus for the LSP

The business case for the LSP identified two critical issues associated with the production and availability of legislation in Tasmania:

- the throughput of the OPC was considered inadequate to meet the increased demand for new or amended legislation; and
- the lack of consolidated legislation was impairing access to the law, increasing the costs of business and government and reducing the effectiveness and standing of the law and Parliament.

The importance of statute law has increased both generally and in Tasmania, with an associated increase in its volume and complexity. Governments are creating policy covering areas previously only covered by common (case) law. While the 1950 reprint of Australian Commonwealth Legislation involved about 5000 pages, a similar exercise in 1982 was estimated to include 150,000 pages (Ward 1982). The trend towards more statute law has not slowed since that time.

These trends, prevalent in most other Australian jurisdictions, were intensified in Tasmania from 1992 to 1996 by the presence of a majority government in parliament. As the governing party had the majority of seats in the house, government initiated legislation was not debated to the same degree in parliament, and the throughput of parliament increased. The Tasmanian OPC had problems keeping pace with this expanded workload. Compared with other Australian Offices of Parliamentary Counsel, the Tasmanian OPC was under-resourced and the standard of the statute book consequentially suffered. It was the poor state of the statute book, plus the perceived need for a greater throughput of legislation which has prompted the push for automatic consolidation.

Access to consolidated legislation was an ongoing problem in Tasmania. Amendment acts were generally not consolidated into the principal acts and anyone wishing to use acts which had since been amended had to manually include these updates. That is, they would physically cut out sections of the amendment act and stick them into the principle act or would correct the principle act by pen. There were three major periodic consolidations of Tasmanian legislation, in 1902, 1936 and 1959. In 1978 a rolling reprint scheme was introduced but fell into disuse due to lack of resources including funds, time and personnel qualified to consolidate and verify the legislation.

Figure 2.1.1: The process of producing Government Bills before the LSP in Tasmania (source: LSP documentation: "Discussion paper - Implementation of Camera Ready Processes for Tasmanian Bills" 27/4/1995)

the principal act was time consuming, prone to error and directly increased the costs of obtaining legal advice or using legislation. Additionally, the manually consolidated versions of the legislation, termed 'paste-ups', were cumbersome, even if they were correct, as illustrated in Figure 2.1.2.

Significant amounts of time and effort were required to interpret the law from such paste-ups and mistakes resulting from this interpretation impinged on the effectiveness of the judicial courts. In one case, the judge and the lawyers from both parties were all working from pasted-up versions of an act, each of which was different and the case was delayed until the actual state of the law could be determined. The professional body of lawyers in the state expressed concern over the state of Tasmanian legislation and its implications for the Tasmanian legal system.

There were also a number of social costs associated with the lack of consolidated legislation which promoted the LSP. Law based on unconsolidated statutes is complex and requires legal expertise to interpret, imposing unnecessary financial burdens on the community. The poor state of the statute book impacted on the effectiveness of the law, as well as the effectiveness and efficiency of the courts. It also impinged on the effectiveness of parliament, with members of both houses expressing concern about the difficulty of interpreting the legislation they were to debate. Amendment legislation was presented separately and it could be difficult for individuals to track how they related to the principal act.

c) The Legislation System Project

Very little computerised technology was used to aid the production of legislation in Tasmania prior to 1993, and advances in information technologies suggested solutions to the identified problems. The project's Business Case Document states:

The OPC has not benefited from improvements in information systems which have increased productivity in other areas of Government activity. Although at its core, drafting is essentially an intellectual and creative task, the process of producing legislation involves many stages which could be significantly improved by automation and access to better information systems (p 2).

The LSP formally aimed to provide:

- A legislation drafting and consolidation system within the OPC;
- A legislation database controlled and maintained by the OPC; and
- A communications network that provides access to the Bill drafting and consolidation system, and the legislation database (Business Requirements Document: pp 11-12).

Most other state jurisdictions used document management systems but, while all State and the Federal governments must cope with the problem of consolidating legislation,

this is the first time automatic consolidation has been attempted. However, the overview of current issues in the production of legislation in Australia below illustrates that the trends promoting the LSP are similar to other jurisdictions.

2.1.2 The production of legislation: Current issues

Good quality legislation can lead to administrative efficiencies, reduced uncertainty when applying the legislation and better administrative decision making (Falkiner 1992 in Lim 1993). Lim (1993) likens the effects of producing poor quality legislation to the high cost of maintaining inadequate software⁴. The House of Representatives Standing Committee on Legal and Constitutional Affairs Report on Clearer Commonwealth Law (1993) suggested a number of longstanding and new issues in drafting also contributing to the quality of the legislation. These common trends directly influenced the development of the LSP and the implementation of the resulting EnAct system and include:

- the recruitment and retention of drafters;
- general calls for legislation that is easier to understand; and
- inadequate access to up-to-date legislation.

a) The recruitment and retention of drafters (ongoing backlog of work)

In most jurisdictions, there is nearly always a backlog of work as drafters have to operate under strict time pressures to meet the policy objectives of the government and individual parliamentarians (Lim 1993). There is also a general shortage of trained people in this area. Drafting is a specialised area of law and it takes considerable time to develop expertise. This is a problem in all Australian jurisdictions and had a significant impact on the LSP.

b) General calls for legislation that is easier to understand

The complexity of legislation is commonly agreed to be an issue and results, not only from the language employed, but also from the nature of the policies being articulated and the process of creating legislation.

According to Hawkes (1994), the complexity of legal language has its roots in the 13th Century, when both the English language and the legal system we know today were being established. At that time English was evolving out of Norman and other influences. Couplings of synonymous words, such as "null and void" or "cease and desist" were used to aid clarity when it was likely that people would be likely to know

⁴ This comparison between the legislation production process and systems development is taken further in Part 5.

only one of the two words and Latin terms were considered more precise than emerging local language. Long sentences were the norm, as, by law, all legislation had to be written without punctuation. Although an English statute of 1850 allowed the inclusion of full stops, the trend of long sentences persists today as novice lawyers are socialised into existing norms (Hawkes 1994).

Plain English drafting promotes the avoidance of long or convoluted sentences or unnecessary concepts and simple and clear organisation of material (Australia 1993: p 5). An ongoing trend over the last twenty years, it began in the United States in the 1970s as part of the broader consumer movement, when people demanded insurance and other contracts that were understandable. In Australia it has been promoted by Law Reform Commissions. Plain English became an important contextual issue in the LSP, as it came to be embedded in the new system.

The complexity of legislation is commonly agreed to be an issue, but the language employed is seen as only one reason for this. Other reasons for complicated legislation include:

- complex policies to be articulated in legislation. Often policies are ill-conceived or poorly thought out. Sometimes the intrinsic complexity in a particular area may require complex legislation;
- the fact that legislation has to be created to be used by multiple audiences (Evans 1996).
- many amendments in a policy area make the corresponding legislation complex; and
- the emergent nature of the drafting process. That is, it is produced on an incremental and iterative basis, with extensive participation. Changes often occur in policy while legislation is being created so causing the legislation to become more complicated. This approach does not aid the efficiency of the process, but is necessary to achieving effective outcomes. This emergent nature of the process became a great topic of discussion during the LSP, as will be illustrated in Part 4;

c) Inadequate access to up-to date legislation.

Access to legislation has been an ongoing issue in all Australian jurisdictions, with the House of Representatives Standing Committee on Legal and Constitutional Affairs Report on Clearer Commonwealth Law (1993) commenting that, "There is little point having clearly drafted legislation if it is difficult for people to gain access to it" (p 180). Legislation not only has to be physically accessible, but also presented in a form that can be comprehended. The problem is that it has generally only been available in an unconsolidated form on paper.

The consolidation of legislation is an important issue in all Australian jurisdictions, with the House of Representatives Standing Committee on Legal and Constitutional Affairs Report on Clearer Commonwealth Law (1993) reporting that this issue was the one concerning most people:

11.39 The Committee received evidence that Commonwealth agencies alone devote considerable resources to maintaining unofficial consolidations of legislation. DSS spends about \$65,000 to produce each unofficial consolidation of the *Social Security Act 1991* which is distributed widely to staff and welfare groups, and expect to produce three consolidations in the 1992-1993 financial year and at least four the following year. The Attorney General's Department keeps a set of paste-ups covering almost all amended titles of Commonwealth Acts and Regulations.

As discussed in section 2.1.1, this issue was one of the primary impetuses for the LSP and is common to other jurisdictions.

The Committee recognised that "good electronic access to legislation is vital particularly in areas where it is not easy to obtain printed legislation" but concluded that,

... electronic access to legislation, however, is no substitute for an adequate system of publishing printed (and reprinted) legislation. The cost of electronic access and the special skills needed to operate systems for electronic access means not everyone will be able to gain access to legislation electronically. Printed legislation is likely to be more accessible for many people for many years to come (1993: p 194).

However, the rapid diffusion of the world wide web over recent years has facilitated a change of views and much Commonwealth legislation is now available over the internet in a periodically consolidated form, as is addressed in the next section.

This overview of common trends and issues affecting legislative drafting in Australia reveal that most of the broad reasons for the LSP are common. This suggests the EnAct system resulting from the LSP project will have significance for other Australian governments but, as illustrated in the next section, to date it is a unique systems development initiative.

2.1.3 Statute law and computers

Computerisation has been considered in relation to statute law primarily in two ways. Firstly, information retrieval systems can aid access to the body of statute of law. Secondly, desktop publishing and other systems can help improve the process of creating legislation. As discussed later, the LSP covers both of these areas though here, as is generally done, they are examined separately. This examination of historical trends in this area sets the scene for the development of the LSP.

a) Accessing statute law: Information Retrieval Systems

Since the 1970s governments and private organisations have been developing and providing legal information systems covering both statute and common law. Here only the historical development of information retrieval systems covering statute law is covered.

The development of information retrieval systems in Australia has been marked by long-standing monopolies. From the early eighties a private company, CLIRS, was given exclusive rights to disseminate Commonwealth and much state legislation electronically. Following a ministerial decision in July 1981, their STATUS system became the nationwide standard for legal information systems in Australia and so alternative approaches were locked out of the market (Ward 1981).

Ward reports this monopoly played a significant role in the developing use of information retrieval systems for legislation over the next fifteen years in Australia. CLIRS retained exclusive rights until 1989/90, with an option to extend to 1993/1994.

However, SCALE did not diffuse as widely or as quickly as anticipated because:

- earlier versions of SCALE systems only had a small amount of information and so were of limited benefit;
- the systems were expensive;
- most users found the system only marginally more convenient than using paper versions of the acts; and
- most who would be likely to use such a system found it extremely difficult and unfriendly to use (Greenleaf et al 1988).

For many of the Tasmanian drafters, this system was their only first hand exposure to legal information systems prior to the LSP and they were not impressed. Hull (1994) commented it was "hellishly hard to use", with no facilities to point and click, for example.

In other words, it requires key-by-key precision that only computer nerds can work (p34).

One of the developers of the system wrote a different interpretation of why the system was not accepted to the degree anticipated:

Irrespective of the merits of a legal computer based system, acceptance has been relatively slow, particularly by those lawyers whose legal research techniques are deeply ingrained. Even if the computer based system is viewed favourably by most, some professionals will probably still continue to avoid the new technology.

Fortunately, this conservative attitude is not so great with the younger lawyers who have been trained to accept technological change... (Ward 1981: p 169).

In other words, according to the systems developers, it was the conservatism of the legal profession rather than the system that was the problem so that the issue was an implementation, rather than one of planning or design. This issue is taken up in Parts 3 and 4.

In the early 1990s, the field of legal information retrieval systems became much broader, with the entrance of new players. Greenleaf (1994) commented that the situation was now becoming more fluid as various governments abandoned efforts to restrict the availability of computerised legislation. The UK LEXIS system was

expanded to include Australian Commonwealth and some state legislation. Various "boutique" databases covering only one or several acts in a particular areas such as taxation or company law were being developed by private organisations and were disseminated by either CD-ROM or floppy disk. Greenleaf commented that the developers of such "boutique" systems often aimed to provide more services than the generic systems, such as SCALE or LEXIS, so pushing developments in the area.

Most providers obtained information from the offices of Parliamentary Counsel in the relevant jurisdictions, but did not add value to that information in any way apart from entering it into an electronic format and disseminating it electronically. In 1994, there was only one database provider which included its own consolidations of legislation (DiskROM's *Consolidated Commonwealth Statutes* CD-ROM) (Greenleaf 1994). A submission to the Standing Committee on Legal and Constitutional Affairs (1993) commented that current CD-ROM services in this area were limited because they "mirrored the deficiencies of a paper-based system" (p 182).

By 1994, most Australian jurisdictions had electronic access to Acts of Parliament, with some states providing access to legislation consolidated on a periodic basis (Greenleaf 1994), as is illustrated in Table 2.1.1. Although the state of play has changed over the last four years, the data from 1994 is displayed here to give an indication of the state of play at the time the LSP was being planned and defined.

Jurisdiction	Systems providing access to the bulk of legislation in the jurisdiction	Access to amendment legislation
Commonwealth	SCALE DiskROM's Commonwealth Statutes and Statutory Rules LawPac CD-ROM: Commonwealth Legislation	Planned to be fully consolidated with historical information on some Acts by mid 1994 Periodic consolidations(1994) Only reprints
Queensland	QLIRS' Legal Retriever	Up-to-date reprints of some high-priority Acts and other unamended principle Acts (1994)
New South Wales	LawPac	Reprints only (1994)
Victoria	Info One LawPac DiskROM	Periodic consolidation Reprints only Periodic consolidations (1994)
Tasmania	LawPac LSP to fill this void	Reprints only
Australian Capital Territory	ACT LawNet	Periodic consolidation
Northern Territory	SCALE	Periodic consolidation

South Australia	SASACT (based on SCALE) LawPac	Periodic consolidations Reprints only (1994)
Western Australia	SWANS	Only available in consolidated form

Table 2.1.1: Information Retrieval Systems for Acts of Parliament in Australian Jurisdictions 1994 (Greenleaf 1994).

The years from 1994 until 1997 have been a time of great change in this field, with the laws of many states and the Commonwealth now being available over the internet or via CD-ROM (Select Databases 1997). By the time the LSP had been completed, most other states or private organisations had bypassed the problem of access to consolidated legislation by periodically producing manually consolidated legislation available in an electronic form, and had widely disseminated their work by publishing it on the web or some other media. In Tasmania, a private organisation produced consolidated legislation on CD-ROM.

These products, though, all rely on manual or periodic consolidation, and so will need constant updating, while the EnAct system resulting from the Legislation System Project enables automatic, or ongoing consolidation. Additionally, periodic consolidation does not easily allow consolidation to a previous point in time. Crimes or incidences are judged according to the state of the law at the time they occurred, so consolidation to a previous point in time is a useful function which periodic consolidation does not easily support. Thus the EnAct system resulting from the LSP is a significant improvement.

b) Use of computers in the production of legislation

As well as aiding access to the law through improved searching tools, computerised information technologies can assist in the production of legislation. In the early days of the LSP project, a business analyst surveyed other states' OPCs to find out what computerised technology they were using. Most were using wordprocessing technology to create the legislation, as illustrated in Table 2.1.2. At the same time, most states have been developing consolidated information retrieval systems but no other state than Tasmania seems to have conceptually link these two areas of computer use. As defined in the LSP, both the provision of access to legislation and the creation of that legislation were linked in the one broader system. Since then, Victoria has embarked on developing a sophisticated document management system to track the progress of draft legislation and provide better access to legislation, but have not included automatic consolidation. Thus, the LSP is a unique systems development project, although the trends promoting it are common.

Jurisdiction	Technology employed for creating legislation (1994)
Commonwealth	Very few drafters use PCs. Most use dictaphones.
Queensland	14/18 drafters create drafts on workstations.
New South Wales	All drafters have a PC but there is no indication as to how they use them.
Victoria	Computers on all desks and the jobs of keyboard staff said to be threatened.
Tasmania	Isolated desktop computers used to type up drafts.
Australian Capital Territory	All drafters work with PCs.
Northern Territory	Computerised since 1986. Drafters started to use system in 1991.
South Australia	About half of the 11 drafters use PCs to draft and edit
Western Australia	"Most" drafters use desktop technology

Table 2.1.2: Survey of use of computers by drafters by LSP analyst (1994)

c) Other applications of computers in legislation drafting

Lim (1993) suggested some other possible applications of information technologies in this area include:

- Group decision support, such as electronic meeting systems, and project management tools to manage the distribution of work and communication between parties;
- Structuring tools to aid the manipulation of the structure of legislation through graphics;
- Electronic thesauri to keep track of terms and their meaning;
- Modelling, where legal expertise is integrated into an expert system. An example of such a system is LEDA (Voermans and Verharen 1993);
- The automated intelligent comparison of legislation. That is, using computers to compare different versions of a document. This includes both different versions of the one act being developed, or different acts from other jurisdictions to compare them; or
- The automatic application of drafting guidelines. In much the same way that spelling and grammar checks can be applied, so to can broader guidelines, such as those for Plain English drafting.

However, Lim noted that:

Despite the wide scope for the application of IT to legislative engineering⁵ there has not been universal development or acceptance of these tools.

⁵ the term Lim employs for a drafter

While emphasising that computers cannot replace human skill and can only play a decision support role, he comments,

In order to understand the failure of legislative engineers to adopt IT to date, it is necessary to appreciate resistance within the legal profession in general. Lawyers have traditionally been resistant to change, specially in their practices. With respect to computers, they have been particularly obstinate.

Like Ward (1981), Lim emphasises that lawyers tend to “resist” changes involving computerisation. The LSP systems developers expressed the same view, but is it entirely fair? Chapter 2.3 and Part 3 will investigate this issue and suggest that these opinions can be traced to an inadequate perception of the process of systems development.

2.1.4 Reflections: The LSP – a unique project induced by common trends

In summary, the drafting of legislation seems to be one of the last information-based professional areas to be affected by computerisation in Australia. Although a very specialised area, it is a crucial one for core government activities. Inefficient or ineffective drafting and inadequate access to legislation can create severe bottlenecks in the workings of government and the application of laws. Computerisation has been seen as one way of overcoming problems in these areas but, until the LSP, legislation drafting and statute information retrieval have been considered only separately in Australia.

The LSP is thus a unique project. Neither the systems developers nor myself have been able to find a comparable project, both technically and procedurally. While there is access to other states' statute law through information retrieval systems and most states' OPCs use desktop publishing in some form or another to aid the production of legislation, this is the first time that these two elements have been combined in one project or system. In other jurisdictions, they are considered separately, although the output of one feeds into the other.

Comparing the LSP with general trends identified above, it becomes obvious that the issues promoting the LSP are similar in most other comparable jurisdictions and could be of relevance to them. As in other jurisdictions, the workload of the Tasmanian OPC is increasing due to general trends towards more statute law with greater complexity. Problems of access of consolidated legislation are widespread. These trends are no different from other jurisdictions, but are exacerbated in Tasmania due to the state's small size. Thus the forces promoting the LSP are no different in other states, but are perhaps more exaggerated.

This description of the broad context provides a background for the LSP. While many of the trends promoting it are common to other jurisdictions, these contextual

elements have had a great impact on the proposed content of the project as well as the process by which the project proceeded. This examination of the broad context of the LSP also indicates the importance of the context to systems development initiatives. The important and fundamental influence of the context is examined throughout the following chapters.

2.2 Conceptualising the organisational context

The immediate context of an information systems development project is its organisational setting. It includes aspects which impact and are impacted by activities involved in systems development projects, but which are not usually considered part of that process. The field of organisational studies is complex and fragmented and one useful way of examining differing perceptions of organisations is to focus on the organisational metaphors employed. As well as introducing relevant issues for developing a model of the organisational context of systems development in Chapter 2.3, and the importance of the context of change, this analysis illustrates that differing perceptions of the organisational context can result in different conceptions of the process.

In order to understand the process by which the LSP unfolded, the specific organisational context needs to be explained. A review of the literature reveals a large number of approaches for analysing the context of systems development. Which one is appropriate for analysing the context of the LSP? What are the possible implications of using each approach? This chapter addresses these questions by reviewing common models and metaphors for analysing the organisational context of systems development.

Common metaphors for organisations include viewing organisations as machines, organisms, political systems or cultural systems (Walsham 1993). A description of these metaphors reveals a number of underlying assumptions one can make about the organisational context of systems development and raises points which are significant to later discussions. This chapter will illustrate that four common models of the organisational context of systems development reflect different organisational metaphors. It will then argue that the model adopted will have a significant influence on how one views the process and possible content of the change. This analysis will also introduce several key points of great relevance to later discussions and provides justification for the organisational model introduced and applied in Chapter 2.3.

2.2.2 Metaphors as perspectives on organisations

The context of an information systems development initiative is its organisational setting. The field of organisation studies is fragmented, with little agreement as to the nature or validity of the explanatory concepts used (Ashmos and Huber 1987; Orlikowski and Robey 1991; Wolfe 1994). Eldridge and Crombie (1974) described the field as "a babble of voices" while Williams (1982) wrote that "the picture provided by the sociology of organisations is as bewildering as any of Kafka's novels"

(in Aungles and Parker 1992: p 55). The wide range of such models is indicative of the diverse state of organisational studies in general.

One appropriate way of looking at organisations is in terms of metaphors (Morgan 1986, Boland and Greenberg 1992, Walsham 1992). Metaphors usefully emphasise that situations can be interpreted in alternative but equally viable ways. The approach has been widely adopted for defining situations and classifying perceptions of them. The recognition of multiple metaphors implies that it is valid to have differing perspectives on the same phenomena.

Walsham (1993) suggested four commonly employed organisational metaphors for organisations. They include viewing organisations as machines, organisms, political systems or cultural systems. Organisational models can be classed according to what metaphor they reflect and this can reveal assumptions and oversights of the model. Each of these organisational metaphors focuses on some elements of an organisation at the expense of others and, if this dissertation is to utilise an appropriate model of the organisational context of systems development, it must identify what kind of organisational model and metaphor is appropriate. This discussion justifies the organisational model introduced and deployed in the next chapter.

Morgan noted that comparing organisations to machines or organisms is very common. Taylor's (1912) principles of scientific management adopts a mechanical metaphor by assuming it is possible to create a routinized, predictable and totally controllable organisation. This mechanistic metaphor underlies much management and systems development literature (Boland and Greenberg 1992).

With a growing recognition of the social aspects of organisations associated with well-known research projects such as the Hawthorn Studies (Mayo 1949), the development of the socio-technical systems approach (Trist 1981) and the contingency approach to change (eg Lawrence and Lorsch 1967), organisations came to be likened to organisms (Walsham 1993). IS literature based on this metaphor include the stage theory of information systems development (Nolan 1979) and strategic grid for information systems planning (McFarlane & McKenny 1983 in Ward and Griffiths 1996). Change is also viewed as a biological phenomenon, with a focus on lifecycles and patterns of predictable growth. Walsham (1993) suggests that prescriptions arising from this literature can be valuable for thinking about issues, but tend to be simplistic when trying to apply them to specific organisations as they ignore the fact that the relationship between organisations and technology is complex and non-deterministic.

Building on the work by Morgan and others, Walsham (1993) describes the manner in which people use metaphors to define their reality and the utility of doing so. After

discussing the advantages and disadvantages of comparing organisations to organisms or machines, Walsham suggests it would be more useful to liken organisations to political and cultural systems to highlight the social nature of organisations. This chapter backs up Walsham's conclusions.

Treating an organisation as a political systems has a long history, with Cyert and March (1963 in Walsham 1993) identifying conflicts of interest as fundamental characteristics of an organisation. Goals, they argue, are established through bargaining and coalition building and the use of authority. However, Walsham (1993) comments that power and politics are rarely mentioned in relation to information systems and suggested this may be because of the myth of rationality in modern organisations, a theme pursued throughout this dissertation.

This view of organisations as political entities is not necessarily a condemnation of organisations as forums of conflict and manipulation. Aristotle's famous phrase, "Man is a political animal" has come to be interpreted often as "People are conniving creatures", yet this was not the original intention:

...man is by nature a political animal; it is in his nature to live in a state [a city, or community]. He who by his nature and not by ill-luck has no city, no state, is either too bad or too good, either sub-human or superhuman (Maddox 1985: p 3).

People are essentially social creatures and, as they interact, they create shared understandings and goals through negotiation. Such mutual learning and compromise does not necessarily imply conflict and cohesion. There is a risk that a political stance focuses on conflicts between groups and individuals as they act in accordance to their different worldviews. Politics is not necessarily negative, but inherent in all human actions. It helps researchers and practitioners recognise that many interactions are between people with different perspectives, interests and power. Thus, it is important to recognise the political nature of organisations here.

Viewing organisations as cultural systems suggests they are patterns of symbolic discourse and action, created and recreated by the people who form them. Subgroups can have subcultures which exhibit distinctive characteristics and ascribe different interpretations to the same events so multiple perspectives must be considered. As a phenomenon created through ongoing interactions, organisations are dynamic entities. The implications for management are that organisations cannot be controlled, though their evolution may be influenced (Walsham 1993). Viewing organisations as political and cultural systems usefully helps us focus on the social interactions which create and recreate organisations and the actions of people involved.

2.2.3 Considering the context of systems development: some approaches

The understanding and management of the organisational context is broadly recognised as an important part of system development projects and there are many models focusing on it. Four common ones include:

- Structured approaches to systems analysis (eg Yourdon 1989, Powers et al 1990)
- Scott Morton et al's (1991) MIT90s model;
- Kling's (1987) description of a specific web model; and
- Checkland's (1988) CATWOE analysis from Soft Systems Methodology.

If the details of these models are examined and compared, we will be able to see that how one views the context can influence how we view the process and the possible content of change projects.

a) Structured systems analysis and design

Structured systems analysis and design is a conscious effort to make systems development more of an engineering discipline than an art and strongly reflects objectivist ontological and epistemological assumptions (Smith 1997). Structured methods are very common, with widely used examples including Structured Systems Analysis and Design Methodology (SSADM), Information Engineering, MERISE and Jackson's Structured Methodology (Avison and Fitzgerald 1995).

These approaches have much in common. For example, they promote three separate views of a system including:

- a time/control view, often represented by a state transition or entity life history diagram;
- a functional view, generally illustrated through data flow diagrams; and
- a data view, often represented through entity relationship diagrams (Bansler and Bødker 1993).

A structured approach thus analyses an organisation in terms of its work processes and data requirements. These are generally described in diagrammatic terms, much like a electrical circuit chart or block diagram. The mechanistic approach to the organisational context is obvious, with little recognition of political or cultural factors. Not surprisingly, given the aims and engineering background of the approach, organisations are typically treated as machines or systems. The diagrammatic approaches used are based on process flow charts, an industrial engineering approach

developed by Taylor (1912) and Frank and Lillian Gilbreth (Bansler and Bødker 1993). As Bansler and Bødker comment,

...the organization is seen as a "machine" designed to perform a given function optimally (in some sense), and work is essentially treated as procedural in nature, involving the workers' execution of a prescribed sequence of steps (p 173).

Although focusing aspects of the organisational context, Kling would describe this approach as a discrete-entity one. Only those aspects that are directly related to the technological system are considered. There is the assumption that the function of the system can be separated from its implementation. Politics is seen as something to be avoided. For example, Yourdon (1989) suggests minimising or even skipping an analysis of the current physical system for "political" reasons and user frustration (in Bansler and Bødker 1993).

The process as well as the context of this approach is viewed as an engineering one. Design is seen as a problem solving process (Bansler and Bødker 1993), starting with a well-defined problems and explicit objectives. Users play a largely passive role by providing information and there is the assumption that the designer is completely rational and objective. Therefore, the content of the proposed changes tend to be ambitious (Kling 1987). In summary, if one adopts a structured approach to analysing the organisational context of systems development, as many systems development do, then the organisational context is likely to be viewed as a mechanical kind of entity and the process of systems development is likely to be viewed as an engineering process of problem solving.

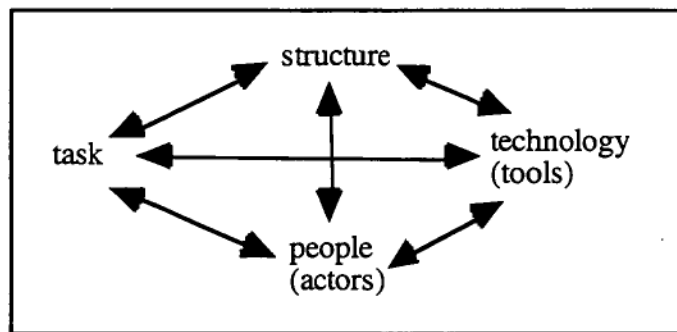


Figure 2.2.1: Leavitt's Diamond (source: Leavitt 1964: p 56)

b) MIT90s / Corporation of the Nineties

The MIT90s model is a graphical model that is widely used to consider the interaction between different parts of the organisation in times of change (McKersie and Walton 1991; Benjamin 1993; Yetton, Craig et al. 1995). It was derived from the work of Leavitt, Whistler and Chandler, and particularly the so-called "Leavitt diamond". Illustrated in Figure 2.2.1, the Leavitt diamond is often referred to in the IS/IT literature to indicate the interrelationship between computer systems development and

non-technical parts of the organisation (eg Keen 1981; Dachouffe and LeSuisse 1989; Noble 1991). The MIT90s model is an elaboration of this diamond model. As Figure 2.2.2 indicates, the model places management processes firmly in the centre of the organisation, redefines "task" as "strategy" to perhaps reflect a growing emphasis on this and "people" was more concisely defined and depersonalised by changing it to "individual roles/ skills". What is meant by each of these categories is not precisely defined. As with all models, the MIT90s framework has developed over time. Widely disseminated by Rockhart and Scott Morton in 1984, it seems to be most well-known because of its role as a unifying framework in the "Corporation of the Nineties" program (Madnick 1991; McKersie and Walton 1991; Scott Morton 1991; Benjamin and Levinson 1993).

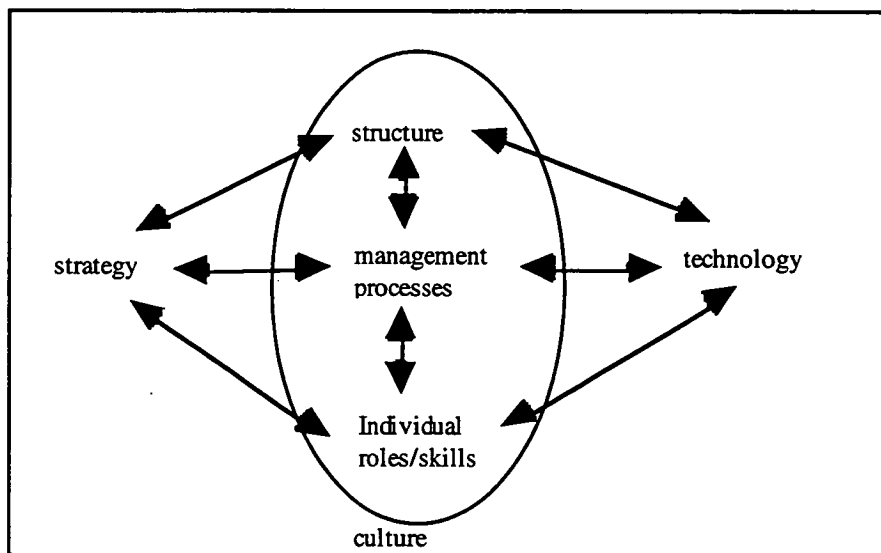


Figure 2.2.2: Corporation of the nineties snapshot model of an organisation (source: Derived from Scott Morton 1991)

Unlike Kling's web model, which assumes that only incremental change is possible, studies using the MIT90s framework tend to assume continuous and major change is important if organisations are to keep up with their environment (Scott Morton 1991). The above framework includes elements that the organisation must keep balanced while change is being managed. Poor implementation of information technology is perceived to result from paying inadequate attention to the individual roles and culture, management processes and structure of the organisation.

Benjamin and Levinson's (1993) model of the change process, which uses the MIT90's model, depicts it as a process of moving an organisation from one state of equilibrium to another. Burrell and Morgan (1979) criticise equilibrium and homeostatic models because such models tend to over-concentrate on internal factors and ignore environmental factors except as a source of disequilibrium (Burrell and Morgan 1979).

The MIT90s model does not fit easily within any of the common metaphors identified above, but its strong focus on growth, sustaining competitive advantage and the emphasis on management prerogatives hints at an organic view of the organisational context to a degree. Management processes are "the glue that holds the organisation together" (Rockhart and Scott Morton 1984), much like an organism's brain or nervous system. "Growth" is an often used term in works referring to the model, as is the idea of evolution, balance and equilibrium. These all suggest a biological metaphor. Change is viewed as natural and something to be nurtured, with an emphasis on growth and renewal. In other words, the MIT90s model hints at an organic metaphor of the organisational context, which suggests systems development is viewed as part of a process of growth and renewal.

c) Kling's web model

Kling effectively argued that information systems need to be deeply embedded in their organisational context if they are to be effective by differentiating between what he termed web and discrete entity (D-E) models (Kling and Scacchi 1982; Kling 1987; 1992; 1994). The basic difference between D-E and Web models is that D-E models do not view the context of information systems as a crucial factor in the success of information systems development while web models do. Much of Kling's writing on web models focuses on their advantages over D-E models.

According to Kling, D-E models commonly underlie systems development in practice. They focus primarily on explicit economic, physical and information processing elements. Historical and social issues are largely ignored. Social arrangements are assumed to be co-operative and resources are assumed to be available when necessary. Information systems are seen as loosely coupled systems that can be broken down and separately analysed and there is the assumption that replacing one piece of equipment with another will produce similar result in all settings. The basic units of analysis are the computing resources and the formal tasks to which they are applied. Behaviour is often seen as being defined by official roles, authority, bureaucracy and goals and so the development and use of computer-based systems considers only those formal elements.

In contrast, the web model is an open system model which focuses on the social relationships and the relationships between people and technology, and the array of activities people do while pursuing tasks. It does not assume that components of a computer-based system can be grafted onto a formal task system in isolation. The situation is the primary unit of analysis, and it varies according to the number of participants, the set of artefacts, the spatial scale and arrangements of activities, the time periods of social activities and primary social processes (1992). That is, web

models use social criteria for defining the situation. Web models do not portray organisations as purely rational bodies, but as "coalitions of shifting interest groups that develop goals by negotiation; the structure of the coalition, its activities and its outcome are strongly influenced by environmental factors" (Scott 1987 in Kling 1992: p 392).

As with other models, Kling's Web model has changed over time⁶. Kling's 1987 web model includes the following elements:

- the social and technological architecture surrounding the existing technology, including work and political processes (ie. workflow and negotiated order), the number of participants, the set of artefacts involved, the spatial scale and arrangement of activities and the time periods involved;
- the social context of the information systems development process;
- the infrastructure of computing development and use, that is, the resources supporting the production processes and computing technologies and the negotiations surrounding them; and
- the history of commitments within and outside the organisation.

Kling (1992: p 367) argues that web models are better than D-E ones, because they give a better account of critical aspects of IS development, such as cost and effectiveness, the speed of the planned change and the integration of the computer system into organisational life. He described D-E models as being analytically simplistic. They tend to only look at economic and technical aspects and so lead to systems developers over-rating their chances of success. Kling argued that, web models offer a more realistic picture of systems development and are likely to make systems development look less exciting, but also less frightening for those involved.

Kling provides several examples of simple situations interpreted using a web model (Kling and Scacchi 1982; Kling 1987; 1992). Sauer's (1993) interpretation of the Mandata project, a large Australian government pay/personnel system, using a web model gives an indication of the depth of analysis that can be involved. Such web models highlight the strongly political nature of systems development and its close relationship with many aspects of the organisational context. Normative web models (eg Kling 1992) are also likely to take more effort than a D-E one and one can easily be overwhelmed by the details. All models are simplifications of reality that are used

⁶ Kling recommended that I concentrate on the 1987 version of the model as the 1992 version had been directed towards systems analysts rather than researchers (Kling, 1995).

to make sense of reality but the web model can be criticised for not producing enough simplification to allow comprehension.

For Kling, change is an incremental, political process involving much negotiation. Referring to Cyert and March (1963), his analysis is openly political and interpretivist. Organisations are seen as the result of ongoing negotiation and change involves a renegotiation of the social relationships which make up an organisation. This approach to change stands in contrast to the previous two models. In viewing the context as overtly political, the process of systems development is portrayed as negotiation and the content the outcome of extensive compromise.

d) SSM- CATWOE analysis

Soft System Methodology (SSM) outlines a process for developing solutions for perceived problems (Checkland 1988; Checkland and Scholes 1990; Checkland and Howell 1993). Checkland believed that, while general systems theory formed a good starting point for analysing problem situations, it is not applicable to human activity systems where there are ill-defined problems. This is because in such situations it can be impossible to answer the questions, "What is the system?" and "What are the system's objectives?". Hence, instead of attributing human systems with goals, Checkland viewed them as having systems of purposeful action by people in social roles. The core purpose of human activity systems is termed the "root definition" and in order to formulate it, Checkland suggested analysing the Customers, Actors, Transformation processes, Weltanschauung, Owners (of the problems, who could stop the transformation processes) and Environmental constraints. This "CATWOE" analysis enables an evaluation of the organisational context.

Checkland saw these systems, or "holons" as he termed them, purely as an epistemological device rather than a description of real world activities. Each holon was developed from a particular perspective and so it is possible and advantageous to collect a number of different holons from differing perspectives. One perspective is a logic based stream of enquiry developed through an investigation of root definitions, while a second looks at the cultural aspects of the situation. The cultural inquiry includes two streams of analysis: social system analysis and an analysis of the political system. The social system analysis focuses on the interactions between roles, norms and values while the political system analysis looks at how power is expressed (Checkland 1990). Although the issue of the observer's subjectivity is not addressed, Checkland's CATWOE analysis does adopt a strongly interpretivist epistemological stance and tends to reflect a cultural metaphor of an organisation.

Understandings of the problem situation lead into a process of change that is likened to cycles of learning, with the above models used as a starting point for

communication and discourse. The process is an iterative one, at least in the initial stages. SSM also does not look at the implementation of change, suggesting the planning of change can be separated from its implementation. The actual actions to solve the problem situation are not discussed and are seen to have arisen out of the previous analysis and, after this has occurred, a new learning cycle begins.

2.2.4 Reflections: The strong connection between the context and process

How we view the context of systems development has significant implications for how we perceive the process of projects such as the LSP. Each of the above models of the organisational context tend to portray the process of developing and implementing information systems in a different manner. Kling's web model suggests only incremental changes are possible, due to the socially negotiated nature of the context while Checkland's CATWOE analysis indicates the process is initially likely to be iterative, though exactly how the changes are to be implemented is not examined. On the other hand, those who use the MIT90s model suggest that large-scale organisational changes associated with the development and implementation of information technologies is not only possible but desirable, while structured approaches simply view the change process as a mechanical process of problem solving. Each of these differing perspectives thus can have a large influence on how we view the process of planning, developing and implementing a system such as EnAct. Hence it is important to adopt an appropriate model of an organisation as the basis for a model of the organisational context of systems development.

Simplistically, but not unreasonably, each of the context models described above tend to portray organisations in different ways, as indicated in Table 2.2.1. How they portray the organisational context greatly influences the way they perceive the process of information systems development and diffusion, and change in general. The very common, mechanistic approach to organisations suggests systems development should be viewed as a process of problem solving as in an engineering discipline. If one adopts an organic metaphor of organisations, the corresponding metaphor for change is one of growth or evolution. Alternatively, if we view organisations as political entities, then the process of systems development and diffusion is likely to be viewed as a process of negotiation. If we perceive organisations as cultural systems, then change can be likened to learning. Although the implied relationships are simplistic, they are no more so than the common attempt to classify models of organisations as mechanistic or organic.

organisational metaphor	systems development context model	process metaphor
mechanistic	Structured development	problem solving
organic	MIT90s	growth/evolution
political	Kling's web model	negotiation
cultural	SSM's CATWOE analysis	learning

Table 2.2.1: The relationship between metaphors used to describe the organisational context and process of organisational change associated with information systems development.

This argument suggests it is crucially important to develop an adequate model of the organisational context of systems development before considering the process. If we are to analyse the organisational context of information systems development, we need to determine what aspects should be considered. Together, these aspects form a model of an organisation. The examination of how context is viewed in various models introduced in this chapter provides a starting point for developing such a model. Which one provides an adequate framework for analysing the context of the LSP and illustrating the relationship between the context and the content and process of systems development?

Any model or theory will focus on certain aspects of a situation, such as the context of the LSP, and ignore others. As Walsham (1993) pointed out, any model is both a way of seeing and a way of not seeing and this is where their value lies. By blocking out extraneous information, they help us make sense of the "buzzing, roaring confusion of reality". Strictly adhering to any of these models in analysing the situation is inevitably going to limit and shape observations and descriptions in some way.

While such existing models have been useful for guiding observations of the LSP, it would be misleading to say the analysis presented here follows one of them. The existing models have provided a sensitising role, by suggesting elements which should be considered. For example, Soft Systems Methodology helps to focus on the differing interpretations that people can have of a situation and also provides tools, such as rich pictures, for describing and analysing these situations and differing perspectives of them. The major strength of Kling's web model is that it illustrates all elements of an organisation are tightly interwoven and that social structures are a result of negotiated action. The organisational change associated with systems development is, by nature, social change, so the more social metaphors are more appropriate. Therefore, this dissertation utilises a more political and cultural perspective. Kling's web model, Checkland's SSM and other similar models form the basis for this framework, as pursued in Chapter 2.3.

In summary, this chapter has:

- illustrated the importance of considering the organisational context when considering the systems development process;
- argued that how the context is viewed will greatly impact on perceptions of the process;
- outlined and critically examined some possible ways of describing the context and process and suggested that political and cultural metaphors of organisations, with the associated process metaphors of negotiation and learning, are more appropriate than other examined models. This approach stands in contrast to the commonly employed structured approach which views the process as a problem solving one.

In addition to illustrating the significance of contextual issues, this chapter has introduced some important contextual issues which should be considered further. This is done in the following chapter.

2.3 The organisational context - the users

This chapter outlines relevant elements of a contextual model of the systems development process and uses this to structure a description of the primary organisational context of the LSP, the Office of Parliamentary Counsel (OPC), as it was near the beginning of the project. Relevant areas to consider include the purpose of the organisation and position in the wider context, the technical / process infrastructure, roles, authority relationships and culture. These elements structured the process of the LSP and help explain why events unfolded in the way they did.

This chapter provides a conceptual model of the organisational context of systems development and uses it to describe the LSP's primary user organisation, the OPC, near the beginning of the project. It will illustrate that many aspects of the organisational context not usually considered in relation to systems development projects can have a great influence on system development initiatives, such as the LSP.

The contextual model described here emerged through the iterative process of observations, analysis of the observations and a consideration of the literature in the area, some of which has been introduced in the previous chapter. Key elements include:

- the purpose of the organisation involved and its position in the wider context,
- the technical/process infrastructure;
- the roles of the people within the organisation
- authority relationships; and
- culture.

The aspects of the organisational context all greatly affected the process by which the LSP was planned, developed and implemented. However, not all the contextual models in the previous chapter covered them all. The commonly employed structured approach only formally acknowledges the purpose of the organisation and focuses only on the technical and process infrastructure. That is, systems developers have largely ignored relevant contextual issues, such as roles, authority relationships and culture. Given the great impact that such elements had on the process by which the LSP unfolded, this is a great oversight. This observation aligns with the conclusions of Kling and Checkland and this model of the organisational context builds on their work by providing a more appropriate picture of the organisational context.

Below, these aspects are explained first generally, and then in relation to the primary user organisation in the LSP, the OPC.

a) The purpose of the organisation and place in the wider context

Any organisation has a defined purpose and a new information system needs to be aligned with it and the overall work processes that support it (Frenzel 1996). The purpose of the organisation is the starting point for considering the context of information systems development projects. Focusing on the purpose of an organisation does not presuppose that organisations have goals, a problematic assumption given that it reifies organisations and dismisses the existence of often conflicting views as to what these goals might be (Aungles and Parker 1992). While goals suggests there is a state an organisation is to obtain, the purpose merely focuses on its reason for existence. A school's purpose is ostensibly to teach while a software development company aims to make money by providing customised computing resources to others.

This purpose of the organisation suggests what the relationship is between those in the organisation and the organisation's context. Pettigrew (1987 in Buchanan and Boddy 1992) divided the context of organisational changes into inner and outer contexts. The inner context includes internal elements of the organisation while the outer one refers to the relationships the organisation and its members have with their wider environment. Buchanan and Boddy suggest that the outer context provides the impetus for any organisational changes but in practice it is difficult to delineate between the two levels of context. The relationship between these levels of context is addressed further in Chapter 4.5.

b) Work processes (technical/ process infrastructure)

The technical and process infrastructure is what Kling termed the "production lattice". Processes are the networks of procedures and activities people in an organisation complete while pursuing their organisation's purpose. Systems development is an example of such a process for many organisations involved in computing, and can be supported by a technical infrastructure of documentation, desk-top computing facilities and CASE tools.

The technical infrastructure supports and enables the process structure. Thus, process and technical changes are often interlinked in both theory and practice. Identifying these processes and suggesting improvements or alternatives utilising new technology is an integral part of systems development and is usually addressed at length in literature in the area (eg Hawryszkiewicz 1994; Shelly, Cashman et al. 1995). With the diffusion of concepts such as Business Process Reengineering (BPR) (eg

Davenport and Short 1990; Hammer 1990; Hammer and Stanton 1995) proclaiming the advantages of reviewing and redesigning business processes and the role of technology as an enabler in this process, the importance of process has come to be generally recognised. However, often consideration of "organisational" or "non-technical" elements only focuses on process issues, and ignores the many other facets of organisations which impact on and are impacted by information systems development.

c) Roles

Essentially, organisations consist of the people involved and their roles in relation to each other. The roles that people play in an organisation are the "building blocks" of that organisation (Carnall 1990). Together, a network of inter-related roles form a working organisation. While the roles people adopt do not always align with the process infrastructure and purpose of the organisation, the focus of this project will be primarily on the ones that do, with observations of people involved in the case studies focusing on professional or work roles and relationships, rather than personal ones.

The concept of roles was adopted from the theatre, the basic idea being that individuals take on a particular role as they interact with others in much the same way that an actor takes on a character role in a drama. These roles can be relational or non-relational (Barley 1990). Non-relational roles refer to the acting out of recurrent activities of a role independently of others while relational roles occur when one interacts with others. Few activities in the process infrastructure are completely independent or relational and Barley suggests that individual's roles are best understood as bundles of relational and non-relational elements, the details of which are negotiated through ongoing interaction. Focusing on how these roles change in a case study of the implementation of CAT scanners, Barley (1990) concluded that technical change could only be associated with social change if these ongoing relationships were influenced. It is thus a great oversight that roles are often not formally examined by systems developers. As will be illustrated below, people's roles greatly influenced how the LSP unfolded.

d) Authority relationships

Within the network of social relationships created as people act out their roles, some individuals have more influence over others. These patterns of influence are the authority relationships which are created and recreated through power. As Giddens terms it, power is the "capacity to achieve outcomes" (1984 in Walsham 1993: p 39). It can be derived from either one's position in the organisational hierarchy or by reference to expert knowledge.

Individuals can gain authority due to their position within an organisation, with a supervisor obviously having more authority than her subordinates, for instance. Pfeffer (1981) argues such power is closely related to organisational hierarchical relationships⁷ which legitimises its exercise (Pfeffer 1981). This close relationship between power and organisational hierarchical relationships is a fundamental aspect of bureaucracy and Weber's (1924) concept of rational legal authority. Rational legal authority involves the creation of legal norms by argument or imposition, with each body of law including a set of abstract rules. A person with authority to enforce these rules holds an "office" and commands issued to others come from the office, rather than the person holding it. The hierarchical authority structures of bureaucratic organisations thus involve a network of official functions bound by rules which are pursued by officials with the competence to administer them. Entry into such a position requires competence but authority is seen to derive from the hierarchical position rather than this expertise.

Sometimes though, people gain authority not because of their formal position but because they have access to particular information due to their position. Secretaries of executive managers have little formal hierarchical authority but can exercise a great deal of informal authority because they have access to information and to the decision-making processes of their managers. Thus the distinction between hierarchical authority and authority derived from expertise is not clear cut.

Pfeffer (1981) argued that power was related more to hierarchical relationships than expertise and access to information, but this has been challenged. With the rise of professional groups and the greater degree of specialisation associated with technocracy, many roles in organisations are specialised ones for people with perceived expertise in a given area. Ilchman (1969 in Burris 1993) suggested that the changing socio-political context of organisations has challenged the assumptions underlying legal-rational bureaucracy promoting what he termed a "rational-productivity" bureaucracy, with a greater emphasis on education, training and experience rather than hierarchical authority relationships. These observations echo Mintzberg's (1979 in Burris 1993) identification of alternatives to bureaucratic hierarchical relationships, such as professional bureaucracies and adhocracies. Burris (1993) suggests these studies are recording a trend from bureaucracies to technocracies.

Technocracy refers to the legitimisation of actions by reference to scientific knowledge or expertise, or rule by experts, with Centeno (1993) terming it an

⁷ Both Pfeffer (1981) and Lacity & Hirschheim (1993) use "structure" to refer to these hierarchical relationships and divisions, a practice not adopted here due to issues raised in Chapter 2.5.

ideology of methods, as opposed to an ideology of outcomes. Technocracy assumes that value assumptions can be ignored, with other perspectives being dismissed as uninformed and that "one best way" is possible and achievable by technical experts (Burris 1993). Technocratic trends had a large impact on the LSP.

Burris noted several characteristics of technocracy, including:

- an increased polarisation between expert and non-expert sectors;
- centralisation is combined with decentralisation in differing configurations. For example, control may be obtained via systemisation rather than personal control;
- skill restructuring;
- expertise as authority; and
- technocratic ideology, which Burris describes as the assumption that technological imperatives have replaced traditional politics in organisational decision making and there is "one best" approach.

Technocratic trends had their origin in the emphasis on science, reason and technical rationality during the Seventeenth Century, a period since termed the Enlightenment. Francis Bacon, for example, promoted a society led by a technical elite who would focus on efficiency and technical order (Burris 1993). Early commentators on technocratic trends tended to be optimistic, predicting that such an approach would lead to benefits for all but, as Burris (1993) notes, these early accounts seem naive. Later commentators on technocratic trends have been concerned about the potential for experts to simply claim authority, with Foucault (1980 in Burris 1993), amongst others, analysing the increasing authoritarianism of knowledge itself. Noting that knowledge is an essential element of power, he suggested the connection between knowledge and power is not new to today's technocracy, but that it has become more pronounced.

Burris (1993) examined changing control structures in work organisations from the 1960s to the 1990s. She observed that bureaucratic structures arose in conjunction with industrialisation and termed it a complementary control structure. Given that production work processes have changed, it is not surprising that there have been corresponding changes from bureaucratic to technocratic control structures. In technocratic organisations, demonstrated or certified expertise tends to become more important than hierarchical position for legitimising power (Burris 1993).

As will be later illustrated, both bureaucratic and technocratic authority relationships were apparent in the LSP study. The primary user organisation can be described as having largely bureaucratic control structures, while the systems developers had a

strong technocratic orientation. These and other crucial differences impacted on the project and its outcomes.

e) Culture

Culture greatly influences how an organisation responds to change and is a focus for many IS researchers (eg Campbell 1984; Sankar 1988; Robey and Rodriguez-Diaz 1989; Cooper 1994; Robey and Azevedo 1994; Tibosch and Heng 1994). The concept of culture is useful because it highlights the importance of social interpretations of cultural artefacts, such as information technology, and recognises that organisations are both enduring and able to change (Robey and Azevedo 1994).

Adopting a symbolic interactionist perspective, Walsham (1993) defines culture as the enactment of shared understandings. It is the process through which reality is constructed as people come to understand situations. It therefore cannot be viewed as attributes of an organisation, but a "living phenomena" through which people create and recreate the world in which they live. As people do not generally understand situations in exactly the same way, organisations, especially large organisations, do not consist of a single culture, but many different and often competing subcultures. This creates what Morgan (1986 in Barrett and Walsham 1995) termed a "mosaic of organizational realities". Counter-cultures, with attributes and values direct contrast to the dominant culture can be a source of conflict in organisations, but also a source of creativity and innovation (Ibid). As Barrett and Walsham (1995) commented, culture is something an organisation *is*, rather than *has*.

An alternative perspective of culture views it as something an organisation has and focuses on organisations' artefacts, values and assumptions (Schein 1985 in Robey and Azevedo 1994). The problem with this approach is that it risks reducing culture to a set of independent variables and tends to promote quantitative approaches which ignore the richness of organisations (Robey and Azevedo 1994). It also tends to down-play the socially constructed nature of organisational reality and the existence of sub-cultures or counter-cultures.

A problem with viewing culture from the symbolic interactionist perspective is that the resulting construct can become too encompassing to be useful (Robey and Azevedo 1994). Focusing on the socially constructed nature of reality defines culture as an extremely nebulous and continually changing construct. Culture becomes a result of people's perceptions and results in physical manifestations such as rituals and myths. Culture itself is difficult to observe and define and it is best to focus on people's expressions of their opinions and viewpoints and any physical manifestations of culture – that is, the precedents and antecedents of culture.

A further problem with the symbolic interactionist perspective is that it ignores the impacts of existing social structures and the need to consider culture alongside structures (Giddens 1979). The two concepts are, in fact, closely linked. Linking the concept of culture with structuration theory, Tibosch and Heng (1994) suggested that culture could be considered both a subjective reality, created by people and their perceptions, and an objective reality, shaping people's actions and perceptions. Culture is, in fact, one of the contextual elements which *structure* people's actions through accepted norms and values, suggesting that the two concepts are synonymous in some respects. It illustrates that the divisions between concepts here (as anywhere) are largely artificial, but are a necessity in order to bring some conceptual order to the tangles of the reality.

In essence, culture can be viewed as a “negotiated reality” (Whiteley 1995) which people experience as an objective reality (Tibosch and Heng 1994) as it becomes institutionalised and objectified as truths (Prasad 1993). Prasad illustrated this process using a case study from the health care sector. She observed that computerisation became associated with professionalism and anthropomorphism as paper systems became viewed as messy and unprofessional and computers were likened to the human mind or a human generally. These are only two of many symbols identified and Prasad concluded that such symbolism can influence actions and have both intended and unintended consequences. Observing the image of computerisation as professionalism aided the implementation of the technology, she suggested that if computerisation were associated with negative symbols for a sustained period there could be problems in systems implementation. Thus cultural structures can greatly influence how systems development unfolds. Prasad's conclusions are backed up by observations of the LSP.

2.3.2 The organisational context of the LSP: The primary users, the OPC

To recapitulate, important elements of the organisational context include: the purpose of the organisation, the technical and process infrastructure, roles, authority structures and culture. While systems developers generally only focus on the first two, they all have a significant impact on projects such as the LSP. As is later argued, consideration of these elements is often simply defined as “implementation problems” or issues of “change management”. This study illustrates, though, that these elements not only influenced the process, but were an integral part of how it unfolded. Thus, if an adequate process model is to be developed, these contextual issues must be considered and their relationship with the process of change examined.

Despite the shortcomings of the structured models formally employed near the beginning of the project, the systems developers were aware that the roles, authority structures and culture could greatly impact on the project, the broad content of which had already been established. The description below of the primary user group, the OPC, illustrates these concerns were justified. As well as “setting the scene” for later descriptions of how the LSP unfolded, this section introduces issues which impacted on the content and process of the proposed changes.

The following chapter will place these concerns in context by examining the context of the systems developers themselves. As far as the systems developers were concerned, though, the primary organisational context to consider was that of the primary users, the OPC. Other organisational units included the Clerks of both Houses of Parliament and the Office of the Leader for the Government in the Legislative Council, but for those creating the EnAct system resulting from the Legislation Systems Project, the Tasmanian Office of Parliamentary Counsel (OPC) was the primary organisational context that had to be considered, and they are the focus of the remainder of this chapter and the dissertation as a whole. This description focuses on the OPC around the time the LSP was being implemented in 1994. Since this time, a number of changes have been observed, and these are discussed in Chapter 3.6.

a) The purpose of the OPC and its place in the wider context.

The OPC is a centralised body of expertise on legislation, providing services to other government and judicial bodies and the public. The office aims to produce accurate, readable and understandable legislation which incorporates agencies' instructions and meets legal requirements and access to consolidated legislation in an affordable and timely manner (22/9/1994). It also provides advice to members of the general public, judiciary or other government agency personnel about the statute book.

Mason (1988) defines the drafting function as,

...the process of taking policy objectives, formulating the legal rules and structures necessary to achieve them, revising those rules and structures in the light of modifications to the policy objectives and, if necessary, revising the objectives in the light, among other things, of the kinds of rules and structures available to implement them (pp 113-114).

In most Australian jurisdictions, an Office of Parliamentary Counsel creates legislation, though there is also the Office of Legislative Drafting in the Commonwealth Government. In other models, drafting expertise is more dispersed with separate agencies being responsible for laws in their jurisdiction. In some Australian jurisdictions, separate agencies prepare subordinate legislation, producing in the Commonwealth "a disturbing variation in the quality" (Mason 1988). It is for this reason, and the highly specialised nature of the work, that drafting tends to be

concentrated in one office as it is in Tasmania. Other cited reasons for centralising the drafting function include that it helps ensure there is a pool of drafting expertise, that the policy formation process does not become confused with the drafting process and it can be a source of independent advice on legislative proposals (Berry 1997). The House of Representatives Standing Committee on Legal and Constitutional Affairs Report on Clearer Commonwealth Law (Australia 1993) suggested this was area of public administration which should not be outsourced. The role of the OPC is thus typical of other jurisdictions in Australia.

Physically, the Office was located near the top of a multi-story building in the state's capital, Hobart. Geographically it was isolated from the OPC's clients, including parliament, most agencies, the government printer and the courts. The offices were designed specifically for the OPC, some features of it including:

- A central typing/administration area where two of the three administrative assistants worked. The entrance was located off this central area and the administrative assistants also acted as receptionists. The main printers were located here, as were time sheets, the keys to the lavatories and other facilities;
- To one side of the entrance was a meeting room large enough for all members of the office, plus a few extra people. A tea room and informal sitting area before the Office's library of legal material were on the other side of the entrance;
- Each drafter had an office around the sides of the building in order to have access to natural light as they worked. Each office had full glass partitions from the rest of the office, deemed necessary as drafting involves a great deal of concentration at times and the noise from others in the office could be intrusive. The glass allowed a degree of supervision and the deputy chief drafter occupied an office off the central administrative area. All drafters had copies of Tasmanian Statutes in their offices, and access to other material in the office's library;
- The Chief Parliamentary Counsel occupied a corner office isolated from the more public areas and the third administrative assistant worked in an alcove outside this office; and
- Proofreaders were assigned a small office off the library, as they had to read out loud.

This description of the physical layout of the office suggests some of the issues which were later to affect the project. These issues included the fact the drafters, including

the chief drafter tended to work in isolation and independently, the office as a whole was somewhat removed from its clients and those with whom it interacted.

The Office itself maintained a high level of security due to the sensitive nature of some legislation before it reached parliament. Access to the office was tightly controlled, with all staff having electronic access keys and visitors being admitted by the administrative staff. As a researcher, I was obliged to sign a document covering non-disclosure of the office's work. Several drafters described the need to maintain the confidentiality of their work, as members of parliament or government bodies did not wish the details of some pieces of legislation to be made public until a certain time. They were highly aware of the political nature of much of their work.

If mistakes occurred, the ramifications could be very public. For example, in June 1995, the front page of the local newspaper reporting that,

Tasmania's 54 MPs voted five times in favour of a Bill before it was pointed out there was a problem with the legislation.

First, the House of Assembly voted three times to support the Pesticides Amendment Bill, not realising that the second and third pages referred to an entirely different piece of legislation, the Veterinary Medicines Amendments Bill.

The Bill then got through two stages in the Legislative Council and was in the committee stages before the chairman of committees... pointed out a major discrepancy in the four-page Bill...

Not only was the title of the legislation changed from one page to another but there was a major flaw with the major point.

The amendment was supposed to change section 10A of the original legislation but in fact the Bill changed section 11A.

Every piece of legislation is supposed to go through a rigorous examination first by the Parliamentary counsel, then by Cabinet, then parliamentary Liberal Party, then at least three specific scrutineers in the House of Assembly and the same procedure in the Legislative Council.

How it got so far without someone noting the whole amendment was a total mess has left everyone worrying (*The Mercury* 24/6/1995).

As well as illustrating the public nature of their work, this incident illustrated the political consequences it could have. There have been numerous debates concerning the viability of having a bicameral system of parliamentary government for the small population of Tasmania and at least one Legislative Councillor used this incident as a rationale for maintaining the house of review. While this problem may not have resulted from an oversight within the OPC, they saw themselves as the “guardians” of the statute book and felt responsibility for its quality. Later in the project, several drafters maintained that their opposition to the project was wholly or partly attributable to a belief that the LSP and the resulting EnAct system would interfere with what they saw as this crucial and fundamental guardianship role

As with any section of the State Service, the office was required to follow the directives of the parliamentary government. They were required to keep up with the

policy programs of the executive government and hence were under a great deal of pressure. One of the drafters stated, for instance,

The OPC is the whipping post for Cabinet. "It's down at the OPC", they complain. They don't see the amount of work we do. Every time a politician opens his mouth it's a day or a week's work for us... They don't understand we're trying to make sense of a policy that might be faulty in the first place or might even be changing as we are drafting (transcripts 3/8/1994).

All policies to be defined in legislation were given a priority by Cabinet and the OPC often found they had much work designated high priority. One drafter suggested this was because politicians are under pressure from interest groups to promote their particular policies and so give the highest priority to their interests, not considering that by doing, so the purpose of the exercise was defeated. Single departments could have a number of areas designated as the highest priority and it was difficult for the Office to determine which needed to be completed first.

- Drafter: There's lots of incredible competing demands from Ministers, the Legislative Council and so forth. They have their "baby" of the time and don't see we are carrying lots of "babies".
- Another drafter: At one time we had a priority one list seven pages long, with five bills on each page. It was simply unrealistic.
- First drafter: One department had four priority one bills- they couldn't even agree themselves (transcripts 20/9/94).

As will be illustrated later, the systems developers were often concerned at the lack of OPC involvement and the possible consequences for the project. The OPC's high workload and the importance they placed on maintaining the statute book on a day-to-day basis helps explain this issue. As the chief drafter commented,

The bottom line is they want legislation. They don't care we are involved in the LSP (transcripts 5/5/1995).

The Tasmanian OPC was certainly understaffed compared to other states, with its seven drafters having to support similar legislative programs to those working in the larger states with more staff. To give an indication of their workload, during the 1992/3 financial year, the Office drafted 150 Bills and 79 Acts were passed while 300 Statutory Rules were drafted and 250 were gazetted. To illustrate the shortage of drafting expertise, one drafter commented that a complex piece of legislation, which had taken much of a year to draft, was passed by Parliament in four days. Members of the Office believed others did not appreciate how much effort went into creating legislation. Depending on the make-up of Parliament, the OPC's workload could change greatly. At different periods they could be spending more time on new legislation or on parliamentary amendments made during debates in either house. As one of the drafters commented,

This government is very busy producing legislation. They weren't for the first year but now they are firing away and the OPC are under pressure. The previous government [a coalition] didn't implement much policy. They were too busy working it out, arguing over what their policy actually meant (transcripts 3/8/1994).

This workload changed during the course of the LSP. During the time of a minority coalition government the OPC had to produce more parliamentary amendments but during the detailed design stage of the LSP, the OPC were spending more time on producing new legislation or amendments to old legislation, rather than parliamentary amendments as a majority government were pushing through their legislative program before an election scheduled the next year. Those involved in the LSP joked they should vote a certain way in the upcoming elections so the OPC would have a reduced workload as the system was implemented.

The systems developers recognised that the users were under a lot of pressure. Crucial stages of the project which involved the users' were initially planned to coincide with the times parliament was not sitting. The implementation of the system, for example, was organised for January 1996 because Parliament does not sit at that time and an election had been scheduled for February. Those involved in the LSP also hoped there would be a reduction in the pressure on the office as the Members of Parliament ran their election campaigns rather than pursued their legislative agenda. The context of the OPC (the wider context) thus greatly impacted on the LSP and how it unfolded.

b) The nature of work processes (technical/ process infrastructure)

The broad processes of drafting and enacting legislation in Tasmania were described in Chapter 2.1. Within the OPC, the creation of draft Bills, amendments and statutory regulations was an iterative process between the drafter, the instructing officer and the support staff, particularly the administrative assistants. The drafters usually obtained instructions from agency representatives or sometimes members of parliament which they attempted to write in a logical and legally binding manner. Before the LSP, drafters generally wrote handwritten drafts, which the administrative assistants typed and returned to the drafters for amendments and corrections. Other support staff were available for tracking cross-references to other pieces of legislation and coordinating the administrative side of producing legislation. The work processes within the office are discussed in more detail in relation to the roles that individuals play within the office and some relevant sections of the systems developers' analysis of these procedures is cited in Appendix 2.

These work processes obviously influenced the shape of the resulting system and thus the process by which the project unfolded. Most significantly, it indicates the drafters had almost no experience with computerised technology near the beginning of the LSP and relied heavily on the administrative staff for typing and other tasks. At least at the beginning of the project, computerisation seemed to be associated with administrative functions, with little relevance to drafters and their work. This brief

description of OPC processes at the beginning of the LSP also indicates the changes associated with the project were significant and represented a major change for the office as a whole and those who worked within it.

c) Roles

The OPC can be divided into two broad categories of roles: drafters and their support staff. The roles are complementary but quite different in nature.

The Tasmanian OPC had seven drafters, including the chief, deputy chief and two assistant drafters. The work of a drafter involved a great deal of concentration as the requirements of the clients and existing legal requirements were incorporated into the structure of new legislation or amendments. The drafters had a further consultative role as they supported parliamentary amendment, or amendments made on the floor of parliament. Hence, a good drafter required an ability to think critically and concisely and to communicate ideas articulately both verbally and in writing. In this way, the role of a drafter could be likened to that of a systems analyst who translates their clients' requirements into legal rather than technical specifications, a comparison pursued further in Part 5.

The role of a drafter is a complex one, involving a great deal of technical and creative skill, as well as the ability to clearly articulate complex issues in a clear and concise manner. Legislation has to be written quickly to meet the political and policy requirements of the elected government but has to also be of a high quality and clearly written. The role is often likened to that of an artist. Hackett-Jones (1988), the South Australian Chief Parliamentary Counsel (CPC), likened legislative drafters to tragic poets, architects, musical composers such as Mozart, film makers, and novelists such as Tolstoy. Such comparisons were also made by some of the Tasmanian drafters. Reflecting the strong language-based background of most drafters, Hackett-Jones (1988) generally used a metaphor to define the role of a drafter:

...the draftsman acts as a kind of midwife and tries by coaxing, cajoling, bullying and bludgeoning to extract the necessary instructions from the clients, often amending the draft on the spot as instructions emerge (p 54).

Hackett-Jones suggested that drafters often try to operate with the instructing officers in much the same way as most lawyers act with their commercial clients. That is, they assume they must obtain clear instructions before acting. He suggested this is unrealistic as policy is still often vague and emerging while the drafters are writing it and perhaps contributing to its content. This separation also assumes that there can be a rigid separation between the form and content of legislation, with the client being responsible for the content and the drafter for its form. Hackett-Jones challenges this

assumption with references to literary criticism and suggests the drafter requires a certain degree of artistic liberty:

Mozart would never have succeeded in composing his *Requiem* if Count Franz von Walsegg had insisted on dictating every note. The same principle applies to legislative drafting (p 56).

The creative nature of legislative drafting is continually emphasised by drafters in Tasmania and elsewhere (eg Evans 1996). This common portrayal of the drafters as artisans and the process of producing legislation as a largely creative activity stands in contrast to the often mechanistic and procedural portrayal of the process by the systems developers, as is illustrated below.

Drafting is a highly specialised area of law and there is a significant shortage of expertise in the area. The drafters commented that it took about eight years of full-time experience to be an experienced drafter, a long lead time which not many are willing to endure. Shortly before the Legislation System Project, the office attempted to employ another moderately experienced drafter but was unable to attract a suitably qualified person. There is a nation-wide shortage of expertise and generally the OPC recruits law graduates with some practical legal experience to train as drafters.

During the LSP, one of the junior drafters left after several years experience to have a baby. She was not offered part-time employment as the management of the office believed drafting was not work that could be completed on a part-time basis due to the complexity of legal issues involved. All the drafters work full-time and most of them had been employed in this highly specialised area for many years. Hence, it was crucial that the new system was acceptable to the drafters. If they did not like the system and use it, the project could not be deemed a success. Stated simply, the drafters could not be sidestepped. They provided a valuable specialised service to the government.

The roles of the support staff were more varied than the drafters' and included an executive officer, a records clerk, three administrative assistants and generally two part-time proof-readers, though this changed during the course of the project.

The executive officer defined his roles as (in order of importance):

- checking the flow of drafts around the Office;
- keeping "marked up" sets of books and maintaining annual volumes of legislation ;
- responsible for reprinting legislation (if there was enough time);
- sending drafts to the printers and maintaining contacts with the printers;
- aiding others in the Office to find information (eg repealed acts) so that the Office could provide advice on legislation;
- Office administrative tasks (fieldnotes 20/9/94).

"Marked up" sets of books were copies of statutes on which amendments to the statute were marked. Both the executive officer and records clerks kept slightly differing

types of marked up statutes which were used to cross-check the other. When queried, these marked up statutes would be used as the definitive source of the law, or the true state of the statute book. The two versions were slightly different in that one could trace when changes had been made to the legislation in only one set.

The records clerk identified his roles as:

- to provide efficient and effective records of the office's work;
- to support the drafters' research requirements;
- to be a member of an operational team in achieving Office objectives and purpose;
- to be pro-active in updating the statute book (statute book is all law as it exists now);
- help provide resources/ work of the OPC to the public (advice re the state of legislation (questioning over whether this is a role now, or will be when the LSP is implemented);
- meet government sector standards and requirements regarding security (fieldnotes 20/9/95).

The administrative assistants performed administrative duties for the drafters and the Office as a whole. In order to complete this work, they not only needed good word processing skills, but excellent grammatical and spelling and an understanding of the format of legislation. When asked to define their work roles, one of them commented, "I'd call it secretarial work, although I don't think it's called that any more. It's too old fashioned". "Just put down 'to do as we are told.'", joked another of them (transcripts 22/9/1996).

The proofreaders were responsible for reading and checking legislation to ensure 100% accuracy and generally worked on a part-time basis. This checking included grammatical and spelling consistency but also against general standards and formats.

The support staff had more varied areas of expertise and roles than the drafters. They all, though, had considerable technical knowledge about the format and structure of legislation and the logistics of producing it. The systems developers found this expertise valuable as they developed the requirements for the new system.

Importantly, many of these administrative tasks were planned to be eliminated, significantly reduced or changed by the new system. Everyone was assured their jobs would not disappear, but that their nature may well change. Given the expertise of the support staff as well as the drafters was essential to produce an adequate system, their attitudes to the project as a whole were crucial.

The distinction between the drafters and support staff is also important and impacted on how the process unfolded. Although on a personal level, many of the drafters and support staff related well to each other, there were also perceived divisions between drafters and the administrative staff. Drafters sometimes felt their expertise differentiated them from their support staff and some of the support staff commented

they sometimes felt their opinions or work were undervalued. The drafters were a little concerned that, with the implementation of the EnAct system, administrative staff would be able to tell the drafters to conform to standards. They felt this impinged on their area of responsibility and would devalue extensive experience and hard-earned expertise. However, on reading an earlier draft of this chapter, one senior drafter asked that I emphasise that the drafters have "...always acknowledged and have been grateful for the meticulous and painstaking way [the support staff] have prepared our work and generally supported the drafters" (17/11/1997). It is important to note that the drafters generally saw themselves as the heart of the OPC and that other staff were there to support them and their work processes. Drafters saw themselves as individually providing services, with the assistance of the administrative staff. This contrasted with the perceptions of the systems developers and, to a degree, the OPC administrative staff, who saw the office as collectively providing a service.

d) Authority relationships

The chief drafter (CPC) was a parliamentary counsel with considerable experience and expertise in drafting and was respected by his staff. Commonly agreed to be a very competent drafter of complex legislation, the parliamentary government often called upon him to create important and time-consuming legislation. This severely impacted on his ability to effectively manage the office and much of this office administration was passed on to the deputy chief. Others in the office sometimes referred to him as "a bit of a hermit". The CPC tended not to be involved in the LSP, at least in the earlier stages, and this caused some problems, as is discussed in Chapter 3.5.

The deputy put a lot of effort into ensuring the office administration was up to standard. Commonly agreed to be a very efficient drafter, the deputy was able to cope with a very high work load and promoted his own standards on others in the office, a move not always appreciated by others. The deputy also felt responsible for maintaining a high throughput in the office to keep up with the extraordinary external pressures. This seemed to cause some problems in the organisation's cultural context, but the deputy chief believed that the high demands placed on the office would not allow a change in management styles (transcripts 3&4/8/94). While not part of the LSP project, this had an impact on the project, as is discussed further in Chapter 3.5. In essence, hierarchical authority conflicted at times with expert authority and this caused problems for the systems developers as they tried to develop functional requirements that everyone would be happy with.

e) Culture

Members of the LSP team often commented on how different the culture of the OPC was compared to their workplace. Most of these comments focused on the hierarchical culture of the OPC, the formalised nature of many of their practices and procedures, a distrust of computerised technology and their perceived conservatism when faced with changing situations. Obviously, these issues all potentially impacted on the LSP.

At the beginning of the LSP, the members of the OPC had very little experience with computerised technology and were not overly receptive to the ideas of the systems developers. One of the systems developers referred to the "traditional nature of the OPC" (1-3/4/97). Some LSP team members were concerned that the LSP could be linked to previous and unsuccessful attempts to introduce computerised technology into the OPC (transcripts 12/8/94). One of the drafters likened himself and his colleagues to dinosaurs and the systems developers to mammals, thus humorously suggesting he felt threatened by the systems developers and the new system. Other drafters and support staff also often suggested they were uncomfortable with the technological changes being promoted. The LSP was introducing significant technological changes but, at least in the early stages, many of the drafters did not seem receptive to these changes. This impacted on how the systems developers approached the process of developing the system.

The drafters were concerned that any technology presented to them should be correct and fit in with their work practices and the systems developers believed they had a very low tolerance for less than perfect technology. One of the consultants involved in the project commented,

Users will have high expectations and this will impose an additional risk for the success of the project (transcripts 8/12/93).

This proved a challenge for the systems developers as they were obliged by senior management to implement a technically innovative system which was likely to have some "teething" problems. This concentration on details could be perhaps due to the drafter's emphasis on the details of legislation, which had to be extremely accurate to withstand scrutiny in courts of law, but made it difficult for the systems developers to show them rough or broad prototypes or concepts. Thus, again contextual elements greatly impacted on the process.

Also of concern to the systems developers was a lack of communication and consensus within the office at this time. The systems developers found it difficult to define requirements when the users themselves could not agree on what they did or wanted. Members of the office did not seem to spend much time talking with each

other. When the business consultant brought in cake to celebrate her birthday, one of the drafters joked it was the first celebration they had had in the office since D-day. This was also echoed in a lack of communication on more work-related issues. Drafters did sometimes discuss their work with another drafter, but this revolved around specific pieces they were writing rather than general approaches. At the beginning of the LSP project, many members of the office complained about a lack of communication about the project and proposed system, as is discussed in Chapter 2.2. Requirement determination meetings sometimes gave drafters the chance to discuss and reflect on the different ways they drafted in a forum in which they could discuss issues on an equal footing. One of the systems developers commented that some of the LSP meetings seemed to be "therapeutic" for the Office, by providing a forum for communication.

Generally those in the Office were extremely intelligent and articulate and were able to express their opinions concisely and clearly, an attribute the systems developers were not always prepared for, nor appreciated. To some degree, this may reflect the adversarial nature of law in Australia. Drafting requires initiative, creativity, intelligence, an ability to think critically about issues and articulate ideas, as well legal skills and knowledge. This aspect of the context proved challenging for the systems developers, whose methodology did not cope well with the OPC's active and articulate response to their work, as discussed further in Chapter 3.2.

These attributes were reflected not only in the drafters' ability to create and amend legislation and bring relevant issues to attention, but in individual work practices, approaches to drafting and social interaction in the Offices. For example, one drafter stated:

We're all one man bands. Put us together and you get a hell of a cacophony (transcripts 25/7/1994).

while another commented,

Anyone who believed in consensus would not have entered the field of legislative drafting. We're all egomaniacs. We are the type of people who argue over the position of commas in a sentence (transcripts 23/8/1995).

In many ways, the culture of the OPC, and the drafters in particular, can be likened to an academic community, which also comprises of individuals who tend to respond critically to issues and work independently. This was an important issue for the systems developers as they tried to work out systems requirements which would meet the needs of all the drafters and the government as a whole. Coming from a team-based approach, the systems developers seemed to find this approach to tasks difficult to comprehend.

There were some cultural differences between the drafters and their support staff and the culture between the two can be contrasted. Drafters tended to critically respond to issues raised while some of the support staff tended not to critically analyse issues to the same degree as the drafters. My impression of meetings the systems developers had with the administrative staff was that they seemed to be far more informal and less confrontational than those held with the drafters. This impression was also echoed by some of the systems developers who were more relaxed and felt less threatened in meetings with just the non-drafting staff. The support staff tended to make fewer comments in general meetings than drafters but were more articulate in meetings without the lawyers, at least at the beginning of the project. The systems developers found working with the drafters challenging, while the drafters intended their often critical responses to the systems developers' work as constructive criticism. These two different perspectives on the same issue are important for later discussions in the next chapter.

2.3.3 Reflections: the importance of the organisational context

In summary, the OPC could be characterised as a reasonably stable professional organisation. It had quite formal and strict authority structures both in terms of hierarchical authority and expertise. Despite changes in the context of the office, internally it was stable with the nature of individuals' roles not changing greatly over the time of the LSP and people have tended to remain in the same roles for long periods. These issues, plus a lack of past experience with computerised technology, contributed to what the systems developers saw as a conservative organisational culture in terms of the office's willingness to openly embrace the new technologies embedded in the LSP. These observations are important. They hint at later problems the project was to face, the fact that current models of the process are inadequate and that an alternative model is required.

The implications of these contextual elements are significant. The highly hierarchical nature of the Office meant that strict protocols had to be followed regarding who was to be informed what and who had a chance to have an input into the systems development project. These decisions were largely outside the control of the systems developers. The drafters had to accept the system or it would not be a success. However, the office was very much an organisation made up of individuals with differing perspectives. Although their input was generally intended to be constructive criticism, their highly articulate dialectic approach to raising issues of concern was not one the systems developers were prepared for nor always appreciated. This reaction, though reflects as much upon the characteristics of the systems developers as much as those of the Office of Parliamentary Counsel.

The organisational model introduced in this chapter provides a framework for introducing some important issues which were to affect the process by which the LSP unfolded. If the context had been analysed using the common structured approach, as introduced in the previous chapter, many of these issues would not have been revealed. Such an analysis would only focus on the technical and process infrastructure and there would be a real risk that the other important elements of the context would be relegated to issues of “resistance” or a need for “change management”. As discussed later, this response would be unwarranted, highly simplistic and a direct result of not adequately considering the context of systems development. Many of the above mentioned contextual issues in the LSP had an impact on the way in which the LSP unfolded and the content of the resulting changes.

Alternative contextual models, such as Kling’s web model and Checkland’s SSM, do investigate these other organisational elements which can impact on the process. They effectively illustrate that change must involve substantial negotiation and learning and the organisational model described and employed above draws heavily on their work. This study backs up their conclusions by illustrating that, due to contextual issues addressed in this and the following chapter, the LSP process can be described at a micro-level as the negotiation of meaning and interest. It supports their arguments that the commonly employed systems development approaches, which were largely utilised by the LSP team, inadequately consider important elements of the organisational context.

In summary, this chapter has:

- provided an organisational model of the context of systems development. The importance of developing an adequate context model has been addressed in the previous chapter. This model covers the purpose of the organisation, the technical and process infrastructure, roles, authority structures and culture. These elements are all crucial for understanding how the process unfolded. This model provides a useful framework for analysing and describing different sections of the context of the LSP, especially the primary users, the OPC and the primary systems developers, the CIPU; and
- described the primary user organisation of the LSP, the Office of Parliamentary Counsel. It illustrated there were some significant contextual issues which probably would not have been formally acknowledged by the common structured approach. Many of these issues were to have a significant input into how the process unfolded, and so this description sets the scene for the “story” of the LSP.

However, it was not only the context of the primary users, the OPC, that influenced this process. The context of the systems developers also had a significant impact on how the process was to unfold, and this is examined in the next chapter.

2.4 The organisational context – the systems developers

The OPC were the primary users of the EnAct system which resulted from the LSP, but they were not the only organisational unit affecting the LSP process. The primary systems developers, the CIPU, had even a larger impact on the process and so should also be examined. Consistent with general trends, the CIPU can be described as having a fluid, largely technocratic, team-based approach, as compared with the OPC's stable, bureaucratic and largely individual work practices. The differing frames of reference of these two units can be compared and contrasted. Simplistically, but not unreasonably, the process of the LSP can be described as one of interaction and negotiation between involved parties, who were operating under the influence of their own organisational contexts.

The systems development literature has generally come to recognise that many aspects of their clients' organisation can impact on the success of the project. However, organisational or non-technical aspects are only generally considered in relation to the proposed users or clients of the system and rarely is the systems developers' context examined. It is taken for granted and thus ignored.

Yet an understanding of the context of systems developers is an important aspect of the context of systems development and can help explain why such projects unfold the way they do. Once we recognise the context of systems developers is likely to be different from that of their clients, there are significant implications for how the process will be viewed. Acting with differing frames of reference, systems developers and their clients are likely to have incongruent interests and meanings which must be resolved. As will be explored in Part 3, this is an important aspect of the micro processes of systems development.

Here the context of systems developers is discussed in section 2.4.1 generally before focusing on the primary systems developers of the LSP, the Corporate Information Projects Unit (CIPU) in section 2.4.2. It will be illustrated that systems developers can work in very different organisational contexts than their clients, and this has implications for the systems development process.

2.4.1 The organisational context of systems developers

Despite the criticisms by Kling and others outlined in Chapter 2.2, the systems development literature is increasingly emphasising the need to understand the context in which the new system will operate. However, it rarely calls for self-reflection on how systems developers themselves operate. As Elkjaer, Flensburg et al. (1991)

comment, there is a lack of consideration on the philosophy and practice of systems development, with systems developers considered as agents in their own right. Yet the context of systems developers can greatly influence the outcome and process of their work. Here the focus is on the context in which systems developers operate according to the themes identified in the previous chapter. This analysis will illustrate that systems developers, such as the LSP's CIPU, operate with differing frames of reference to their clients. This gulf explains why the process unfolded in the way it did to a large degree. Before focusing on the CIPU, it is worth examining the context of systems developers generally. This examination will reveal that the CIPU are typical of systems developers generally in many respects.

a) Purpose and place in wider context

The purpose of systems developers is to aid the development of information systems on behalf of other people. An information system includes artefacts and procedures people use to access and manipulate data and information. Such artefacts include alphabets, books and other forms of technology but the information systems referred to in the literature on systems development generally involve at least some computerisation. This is largely due to the increasingly ubiquitous nature of computing and communications technology but can also be attributed to the technocratic nature of much modern management generally and information systems in particular. Additionally, due to the history of the field, most systems developers have a strong technical background and tend to assume technical solutions (Clegg 1996). This can stand in contrast to their clients, who are probably more focused on their core area of business.

b) Nature of work processes (technical/ process infrastructure)

The technical/ process infrastructure of systems development is the focus on this dissertation and will be examined in depth in Parts 3 and 4. Essentially, there are two basic work processes involved in systems development: systems analysis and design and project management. Systems analysis and design activities basically focus on the content of the proposed system while project management focuses on the process of achieving them.

c) Roles

Systems developers are viewed as experts in systems development and/or project management. They may sometimes only help the potential users of a system to develop their own solutions but generally they aim to produce systems on behalf of others who will use or be clients of the system. Systems developers can include:

- people employed by the organisation who are responsible for systems development activities, whether they do it themselves or contract it out;
- contractors who create systems or parts of them on behalf of the internal systems developers;
- consultants who provide expertise on particular issues or who review the activities of others.

All these different types of systems developers can greatly influence the outcome and process of an information systems project such as the LSP.

Individually, systems developers can take on a number of different roles. A *systems analyst's* main role is to examine and understand the context in which the new system will be used, suggest ways the current system can be improved and to document both the current system and possible future improvements. Sometimes a systems analyst is called a business analyst to emphasise a client or organisational focus. A *project manager* is responsible for achieving the outputs and facilitating the outcomes of a systems development project⁸. His or her role involves planning and coordinating people and resources so that its process and outputs and outcomes are achieved within the planned time and with the agreed level of resources. Other systems development roles include: programmers; technical support officers; project directors, (ie senior project managers who oversee one or more projects but are not directly responsible for managing that project); trainers; systems documenters; and so forth. Depending on their nature, specific projects may require people to take on other roles.

d) Authority relationships

Systems developers generally derive authority from their expertise in given areas, rather than from their position within an organisation. Within an organisational hierarchy, they tend not to wield a great deal of power, so there is a need for an executive champion in systems development. An executive champion is someone with recognised hierarchical authority who publicly supports the project and thus legitimises the actions of the systems developers. The actions of systems developers are legitimised because of their perceived expertise in systems development generally or a particular aspect of it. By referring to the expertise of specialised systems developers, the systems development literature delegitimises end user computing activities. As will be illustrated later, the LSP systems developers were typical in this respect.

⁸ see Chapter 3.5 for an explanation of outputs and outcomes.

Formal authority structures for systems developers tend to be based on the perceived expertise people have when acting in particular roles. A project manager is seen to have expertise in coordinating projects and their role is to coordinate the other experts on the team, rather than control or direct them, for example. Thus formal authority structures in IS/IT organisations tends to be technocratic, rather than bureaucratic as it is based on perceived expertise rather than positions within the organisational hierarchy.

Technocratic ideas have been interpreted into management practices through the development of what is commonly termed scientific management, managerial rationalism or perhaps industrial engineering. This movement is commonly said to begin with Taylor (1912), who redefined management from an art to a science, based on clearly defined laws, rules and principles. Believing that scientific methods are generally too complex for most workers to understand, expert managers were required to run the organisation in much the same way that engineers would maintain a machine. For Taylor, scientific management was not only an issue of promoting raising productivity, but a moral one, as he saw it as aiding price reduction and wage increases generally and the elimination of politics.

Taylor is generally considered to be the founder of what Burrell and Morgan (1979) term the classical school of management, with other noted contributors being Fayol (1916) and Weber (1924). Burrell and Morgan (1979) describe this school as being extremely objectivist, with organisational elements being treated as if they were natural phenomena which could be systematically investigated. The role of management involves the identification and application of fundamental laws and rules of operation, while individuals as a whole are seen as passive entities, their behaviour governed by the structures in which they act.

Although challenged on a number of fronts (eg by Mayo and the Humanist school of management (Burrell and Morgan 1979)), Taylorism and scientific management is still prelevant in management practices and theory today, illustrating that the underlying assumptions outlined in Part 1 also influence normative actions of systems developers such as those involved in the LSP. For example, Davenport and Short (1990) describe the link between information technology and business process reengineering as being based on Taylorist principles, though commented that this approach is now referred to as industrial *engineering*, a term reflecting its objectivist assumptions. This approach is termed rationalistic management by its detractors, who argue that it is not always rational as it only swaps of rationality of ends for rationality of means (Mintzberg 1994). It is strongly linked with technocracy, as introduced in the previous chapter.

Burris (1993) noted that technocracy is most prevalent in workplaces centered around computerised technology and suggested the correlation resulted from two issues. Firstly, high-tech and highly automated organisations tend to operate in highly competitive markets. Innovation aids competitiveness and the people in such organisations may seek technocracy as an alternative to bureaucracy. Secondly, the implementation of computerised technologies presents opportunities for social restructuring, with some arguing that the advantages of such advanced technologies cannot be realised without such changes. It may also be a result of the technical orientation of many of those involved with computing in organisations. The LSP system developers, in common with much of the literature in this area, were overtly influenced by these trends.

e) Culture

It is difficult to find empirical studies into the culture of systems developers, possibly because it is often taken for granted, but it can be viewed as broadly consistent with this technocratic approach. Several researchers have investigated cultural conflicts between systems developers and their clients (eg Hirschheim, Klein et al. 1991; Romm, Pliskin et al. 1991; 1995). However, the actual cultural attributes of systems developers themselves are rarely examined. Conceptual analyses suggest that most systems developers adopt a functionalist approach, emphasising order and objectivism. Epistemologically this stance is linked with positivism and ontologically with realism and is consistent with managerial rationalism (Hirschheim and Klein 1989; 1992a ; 1992b ; Hirschheim, Klein et al. 1994). The culture of most systems developers can thus be described as strongly technocratic reflecting objectivist assumptions, and this had a great impact on how they perceived the process, as pursued further in Part 4.

2.4.2 The LSP systems developers

The State Service's Corporate Information Project's Unit (CIPU) was responsible for the LSP and are termed here the *in-house systems developers* while those CIPU staff members who were specifically responsible for the LSP are collectively referred to as the *LSP team*. The technical development of the resulting EnAct system was largely contracted out to a commercial organisation, here termed the *technical systems contractors*. Other external people were employed throughout the project to review the project, provide specialised expertise or simply more resources at critical times and one of the primary roles of the CIPU was to manage these external parties. Here the focus is on the CIPU, as the key systems development body involved in the LSP. External system developers were outside the scope of this research project and are not described here. This analysis will illustrate the CIPU acted in a quite different context

to their clients and had quite different interpretations of the situation and what actions were required. It is these distinct interpretations which largely explain how the LSP unfolded.

The CIPU can be characterised as a relatively dynamic but fluid group. Like the OPC, it consisted primarily of people with expertise in given areas, but the nature of the expertise required changed often and CIPU staff were expected to gain knowledge in different areas. The rapidly changing nature of the field in which the unit specialised and the specific work loads meant a high turnover of people involved in the unit with those who were involved in the CIPU for an extended period often changed roles. In order to offset this instability and back up recently gained expertise, the CIPU used formal methodologies and models to guide the actions of people faced with often novel situations.

a) The purpose of the organisation and place in the wider context

The Corporate Information Projects Unit was responsible for facilitating the management of information systems projects in the Tasmanian State Service, its formal goal being at the beginning of the LSP,

...to facilitate the implementation of State Service information projects through the application of best practice standards to:

- the review, analysis, design, development, and implementation of systems; and
- project management, quality management, and quality audit consultancy. " (CIPU Annual Report 26/2/97).

The types of projects managed by the unit differed greatly, but all could be classified as information systems projects. Some of these involved the in-house development of such systems but the unit aimed to minimise risk by either outsourcing development work or producing systems based on pre-developed solutions. The unit focused on aiding the analysis, design, development and implementation of systems, project management and quality management consulting, aiming to apply international standards of best practice in these areas. The unit achieved ISO 9002 certification while they were involved with the LSP and supported procedures associated with this standard through a formal quality management system.

The workload of the unit was highly variable, depending on the number of projects it was involved with at a time. Only six people were employed permanently, but this number increased dramatically as projects required more resources. Compared with the OPC, there was a high turn over of staff, with many only being employed for short term contracts. By the end of my involvement in the project, none of the original LSP team members were still with the CIPU. This stands in radical contrast to the OPC, some of whom viewed this rapidly changing staff profile as indicative of a lack of

commitment by the individuals involved. It aptly illustrates the gulf between the two primary units involved in the LSP.

b) The nature of work processes (technical/ process infrastructure)

The work processes of the unit were highly variable, but some consistency was maintained via the use of management tools and techniques, formal methodologies and quality assurance procedures. These provided some external standards through which to compare and improve their work practices and illustrated to auditors they completed tasks to externally recognisable standards. The nature of the CIPU's work processes is not so much part of the context of the systems development process, but the process itself, and is discussed in more detail throughout this dissertation.

c) Roles

The roles of those in the CIPU are more difficult to define than those in the OPC due to the changeable nature of the unit's work processes. These changes are difficult to track and reflect the fluid nature of the unit and the industry in general. The main players and their roles include the following:

- The original project manager of the LSP was also the unit's leader. At the same time as the project was becoming more established, funding was being obtained from cabinet, and the unit was becoming involved in a number of other projects. The project management role was passed onto the then business analyst and the unit's leader was given the role of project director and still gave advice to the new project manager and others in the project team.
- The business analyst was employed originally on a shorter term basis to develop the requirements for the system. This person held the position of project manager for most of the project but when the project director retired in the final stages of the project, she took over his role as project director and unit chief until a permanent employee was appointed. She left the unit just after the LSP was formally accepted by the OPC.
- The new project manager's previous analyst role was taken on by a new member of the unit. He joined the project part-way through the tendering process and was involved in tasks including camera ready procedures and acceptance testing. Being only employed on a short term contract, the analyst left the team part-way through the acceptance testing stage for a new job when the project was delayed.
- This business analyst was replaced by the permanent appointment of one of the consultants who had extensive experience of the project. When the then project

manager took over from the project leader, this ex-consultant became project manager for the final stages of the project.

- The project managers all recognised that user relations were a key issue for the success of the project but felt they neither had the time nor the expertise to concentrate on them. A business consultant came to be employed on a contractual basis and her role was to facilitate communication between the systems developers and the users after the requirements determination stage and coordinate the users' training. She also completed many miscellaneous tasks the CIPU or OPC did not have the resources to concentrate upon and replaced the project manager when she was on leave. It is significant the CIPU did not initially view user relations issues as a core part of their role and contracted it out. Reflecting their strong technical background, it indicated a lack of acknowledgment of the centrality of user relations to the process of systems development.
- The unit's technical support person gave advice to the OPC on issues such as the upgrading of word processing technology and the purchase of new printers. While not officially part of the LSP, the systems developers believed this was an important way of securing the user's trust and enabling the systems developers access to the Office. The technical support person at the start of the project left during the tender stage and the new technical support person was employed specifically to support the OPC at least part of the time. This person also left before the implementation of the EnAct system, but another technical support person was seconded from another agency for the latter periods of acceptance testing and to provide support when the system was implemented.
- External systems contractors were employed to build some of the technical aspects of the system, as discussed in the following chapters and consultants were widely employed. These contractors and consultants provided specialised and expert advice as well as an internal perspective which was used to influence actions. Their influence on the LSP is discussed in more detail in Chapter 3.5.

d) Formal authority structures

The unit was organised into teams concentrating on different projects, with some people being involved in multiple projects, depending on their areas of expertise. Coming from such a background, the LSP team members found it difficult to understand the authority structures and individual approach of the OPC. This again provides a rationale for explaining how the LSP process unfolded.

These trends are reflected in the formal management approach adopted by the unit, which they termed a quality management system. The CIPU at least partly derived

this from Jenner (1993) and the approach was highly documented and developed within the unit. Jenner describes it as a formalised system of management which is aligned with ISO 9000 standards. The CIPU obtained ISO 9000 certification while they were involved in the LSP and put a great deal of effort into maintaining it.

This management approach claims to be client focused, team based and democratic rather than dictatorial and rigidly bureaucratic.

Formalised management systems can and should provide an enriching environment within which people and organisations can flourish. There is no conflict between well-designed formalised systems and a creative, open approach which is small, fast moving, flexible, fun and operates with minimal paperwork (Jenner 1993: p 4).

The unit's director described it as the cornerstone of their management philosophy. The copy of Jenner (1993) which he provided me with describes the role of a manager thus:

Your teams should be set up with no strict management structure. Your role is to support, encourage, train and participate with your team until you find people naturally adopting responsibility for running meetings, communicating with people and groups outside the team, taking charge of when things need to be done, looking after the social needs of the team, providing expertise and advice and looking after the development and improvement of the team's processes.

Compared to the OPC, the management processes seemed far more democratic. In regard to the LSP, the unit's leader took on the role of a mentor, providing advice as well as leadership. The management style of the CIPU was thus aligned with the technocratic trends noted by Burris (1993) and stands in contrast to the largely bureaucratic approach of the OPC.

e) Culture

With its focus on formal project management tools, methodologies and so forth, the unit's culture could be described as being overtly technocratic, but also relatively open to new ideas and concepts. There also seemed to be a greater degree of communication between members of the unit in the early stages of the project as compared with the OPC. They regularly had morning tea together, for example. The unit had a very strong team-based approach to work, with all outputs from the office being checked by other members of the group so that the unit as a whole could take on responsibility for them.

Although the unit was promoted as aiming to meet business rather than technical needs, almost all members had a technical background and this was reflected in the unit's technocratic rather than bureaucratic culture. The people in the unit were generally favourably disposed towards computerised technology, had extensive experience using it and tended to have training and education in computing or related areas.

As well as actively promoting staff education opportunities and using commercial methodologies and tools, the unit extensively employed consultants. The management style of the CIPU promoted an emphasis on external review, which helped facilitate my involvement in the project as they sought constructive observation. When creating the functional requirement document for the project, they informed the steering committee they were seeking review "from as many credible organisations as possible", for instance (transcripts 28/6/94). The CIPU also promoted more external review of their work, while members of the OPC often commented that outsiders could not understand the nature of their work. The contrast between the two groups is dramatic and had a very big impact on the LSP.

2.4.3 Reflections: Differing frames of reference and their impact on the process of change

Observations of the CIPU suggest the unit was aligned with general trends in the literature on systems development. As much as was feasible, the CIPU aimed to adopt recognised standards of best practice. This suggests that the observations made here could have implications for others.

When I started observing the LSP, the CIPU and their culture seemed outside the scope of my project. However, it has become increasingly obvious that it was an important area. Observations of the unit backed up by their comments on the OPC and observations by OPC staff members suggest that the CIPU was far more democratic, had more fluid work practices and hierarchy, promoted a team based approach to work and was characterised by more communication amongst staff members. Particularly, the people in the CIPU had a more positive approach to technological change. OPC staff members commented on the fluidity of people in the CIPU and their technical focus. These differences lead to conflicts of interest and meaning which needed to be addressed throughout the LSP.

In summary, the context of the systems developers can be contrasted against that of the systems developers. Although both part of the same agency in the Tasmanian Government, the CIPU and OPC exhibit quite different characteristics. The OPC can be characterised as a stable bureaucratic organisation with individuals prepared to take a dialectical approach to issues, while the CIPU can be described as a dynamic and overtly technocratic organisation. These differences have great implications for how the process unfolded, as will be illustrated in Parts 3 and 4.

The CIPU could be classed as a technocratic organisation, as compared to the more bureaucratic OPC. There was a team-based approach, which contrasted strongly with the individual approach which was dominant in the OPC. The roles of people within

the CIPU were far more fluid and characterised by a far greater movement of people in and out of these roles.

Significantly, the description above indicates the CIPU had quite different frames of reference to the OPC. Orlikowski and Gash (1994) illustrated how developers' and users' differing frames of reference greatly influenced the process and outcomes of development. Building on work by Gioia (1986), they defined frames of reference as the "repertoire of tacit knowledge that is used to impose structure upon, and impart meaning to, otherwise ambiguous social and situational information to facilitate understanding" (p 176). They describe the difference between groups' frames of reference in terms of their congruity and illustrated that, in their case study, the developers and users had significantly different views of the technology. Previous research had suggested that people who work together tend to share assumptions, knowledge and expectations and social interaction and negotiation created opportunities for developing congruent frames of reference and exchanging different points of view.

These different frames of reference result in conflicts of interest and meaning. Hirschheim (1985) suggested that major technological changes will cause conflicts due to different goals and needs, a theme he and others pursue while illustrating systems development as social interaction (Hirschheim, Klein et al. 1991). Information systems development is basically a process of social interaction between people operating in differing contexts or having different perceptions of the context. Knights and Murray (1990 in Coombs et al 1992) observed cultural conflicts between users and developers in a financial services company. Systems developers and their users are likely to have incongruent frames of reference which may need to be resolved if a project is to be judged a success (Romm, Pliskin et al. 1991; Orlikowski and Gash 1994).

The contrast between the OPC and CIPU is quite striking and, while improved, differences between them were not altogether resolved by the time the system was implemented. It would be simplistic and misguided to say that one was forward looking while the other was regressive, though this was a view occasionally expressed by some of the systems developers. The nature of their work processes were quite different, as was their broader organisational context, the roles people played in each of the organisations and the formal authority structures. Conflicts of both meaning and interest are inevitable when such groups have to work together. Essentially, systems development involves the interaction between different frames of reference and compromises and agreements between them (Hirschheim, Klein et al. 1991). Some specific examples of differing frames of reference are listed in Table 2.4.1.

Date	Incident illustrating differing or conflicting frames of reference
12/5/94	LSP Status Report: "The working relationship between the CIPU and the OPC is not at a close enough level to provide each group with an understanding of each other's needs".
9/12/94 and 14/12/94	Drafters had significant problems understanding how automatic cross referencing worked conceptually and systems developers had trouble explaining it in a way which they understood.
14/9/94	When a user developed a prototype, the CPC commented it was better coming from a drafter. Seemed to be significant that one of external systems developers had a background combining computing and law.
22/2/95	OPC people had trouble with much of the jargon the systems developers employed. The technical systems contractors were told to try and avoid jargon as much as possible.
22/6/95	During detailed discussions about amendment wordings – the Chief Drafter (CPC) commented to the project manager: "Hope you are finding this difficult, like I find some of your stuff a lot of the time".
30/8/95	CPC to one of the systems developers: "I know you people like diagrams"
24/11/95	During quality assurance reviews of manuals for camera ready processes, one of the administrative assistants comments that they would not use the term "execute a program" as the document stated. They would "start" or "open" it.
17/8/1997	Notes on draft thesis by drafter: "I agree with your comments about the fundamental differences and approach between the OPC and CIPU. These two Agencies are, as you point out, totally different, both as to the personalities involved and as to the disciplines they represent. We simply speak different languages".

Table 2.4.1: Some examples of differing frames of reference during the LSP

People's perceptions of the contextual situation depends on the position from which they view it, their experience, background and personal attributes. For the chief drafter, for example, the negative impact of the LSP on the work output of the office during the process of implementation was a significant issue and greatly affected his input into the process of development through his role as a member of the steering committee and a key user. For other users, who were not focused on the OPC's relationship with its clients, this concern was translated more into a fear that the development of the system would require them to complete vast amounts of work in order to keep up. The context, structure or institutional properties do not affect human actions *en masse*, but do according to how people interpret them. Different people interpret them differently, so their actions in response to what may be the same stimuli might be different because they perceive it differently. As one of the drafters stated during the technical systems tendering process:

Whoever gets this tender, they need to understand our world as much as we have to understand their world (transcripts 7/3/95).

The process of achieving this understanding is investigated in Part 3.

In addition to "setting the scene" for examining the process by which the LSP unfolded, this and previous chapters have made some important points. They have

illustrated that the LSP systems developers' frame of reference was quite different to that of the primary users, a point taken further in Part 3. The previous chapter looked at the organisational context of the system's primary users, and this suggested that aspects of the context not normally considered by systems developers would have a great impact on the content and process of the proposed changes and so should be managed. After examining the context of the LSP systems developers themselves, though, the situation becomes much more complex. As discussed in Part 3, the gulf between the CIPU and OPC's frames of reference explains why the process did not unfold in a way the systems developers could control and suggests that, at a macro level, the process is one of negotiation. This stands in contrast to the majority of the literature on systems development, as is addressed in Part 3. It begins to illustrate which the commonly used models of the systems development process are unsatisfactory, do not cope with the different perceptions that participants have of the situation nor the ongoing influence of the context.

2.5 Reflections: Developing a framework for examining systems development and its environmental context – The relationship between Giddens' structuration theory and Pettigrew's contextualist analysis

The previous chapters have made two important points. Firstly, the context has an ongoing and important influence on the process of systems planning, development and implementation. Secondly, participants' differing frames of reference will have a great impact on the content and process of a systems development initiative such as the LSP. Both these points illustrate the closely intertwined nature of the context, content and process of systems development. Parts 3 and 4 will illustrate that the normative conceptions of the process employed by the LSP systems developers do not generally reflect this relationship and so do not adequately reflect the way projects, such as the LSP, unfold in reality.

This dissertation suggests an multi-level process model which does. Before examining and justifying this model in detail and illustrating it with reference to the LSP, it is necessary to step back and introduce the model abstractly. This chapter introduces this alternative process model by providing some theoretical background and considering its relevance to systems development initiatives.

Giddens' (1979) structuration theory and Pettigrew's (1985) contextualist analysis have both been influential frameworks for examining the close relationship between information systems initiatives and their organisational environment. Here the relationship between these two theories is examined and it will be argued that they are largely, but not wholly equivalent. This comparison has important implications for how the process of systems development can and should be perceived.

Particularly, it suggests that the process should be viewed at multiple levels of abstraction. Giddens' three levels of temporality provide a framework for this multi-level model of the process which is articulated in the remainder of the dissertation.

The previous chapters have pointed out the context of systems development has a great influence on the content and process of such developments. The following chapters will illustrate that commonly employed process models, such as those employed in the LSP, do not adequately reflect this and can cause problems. This dissertation provides an alternative model which aims to overcome these problems. Before explaining, elaborating, illustrating and justifying it through a detailed examination of the LSP and related issues in the following chapters, the model is introduced and justified in theoretical terms here.

2.5.1 The dynamic nature of context

The normative literature on systems development emphasises the need to understand the "problem situation" before designing an information system. Analysis techniques to aid this, such as Checkland and Scholes' (1990) Soft Systems Methodology, can be sophisticated and well-suited to this purpose, but they do not indicate how the context not only influences systems development at the beginning of the process, but throughout it. Generally in systems development projects further changes required are managed through change control procedures and will result in a new systems development lifecycle when the system becomes obsolete and difficult to further "maintain".

Truex (1993) suggested that this approach is based on the assumption that the organisational context of systems development is made up of stable structures and criticises it for not recognising that organisations are emergent systems. That is, social systems do not follow fixed patterns, but are continually being recreated and never obtain a stable state. Truex suggests the implication for systems development is that the context should be considered throughout the process of development. However, most only examine it at the beginning of the process, and relegate further influences to "implementation issues" or issues of change management or resistance. This dissertation builds upon Truex's work by suggesting this emergent nature of organisations can be reflected in an emergent model of the change process.

This emergent nature of change in organisations has been extensively addressed by authors such as Barley and Orlikowski. Focusing on the processes of institutionalisation and structuration, Barley and Tolbert (1997) examined the links between action and institutional elements. In essence, they illustrated structuration in action over time as organisations change, by describing the recursive relationship between these two elements. On the same topic at an empirical level, Orlikowski (1996, Orlikowski and Hofman 1997) illustrated how the implementation and improvement of groupware technology in a large software company involved a number of organisational transformations over time. These changes occurred at both the institutional level and the level of everyday actions and changes in each were influenced by and had an influence on the other. By describing the closely intertwined nature of the relationship between organisational structures, they have also illustrated the close relationship between the process and content of change. In Orlikowski's terms, the transformations were situated in their context. Such studies are of great relevance to the themes addressed in this dissertation and are addressed further in Chapter 4.5.

2.5.2 Context and structure

These studies by Orlikowski, Barley and Truex illustrate the very close relationship between the context and the content and process of change (Pettigrew 1985), and the intertwined nature of institutional elements and the actions of those involved (Giddens 1989). Walsham (1993) equates Pettigrew's definition of context with Giddens' concept of structure, and Pettigrew's process with Giddens' concept of human actions (p 69). Here we examine this in more detail and conclude that, while context is equivalent to structure broadly, human action is only partially equivalent to process. This observation will have important implications for how the process of systems planning, development and implementation should be viewed.

The relationship between changes in information technology and the organisational structures in which such changes take place is widely considered to be a crucial issue and has been the focus for much research in this area (eg Borum and Christianssen 1993; Allen and Hauptman 1994). The inclusion of "structure" in both the MIT90s model and its predecessor, the Leavitt diamond (Keen 1981, Scott Morton 1991), and the popularity of such models reflects the perceived importance of the concept.

Unfortunately, though, the term "structure" is often not well defined and is open to multiple interpretations (Barley 1990). Here structure is defined as anything which structures human actions, so recognising that structure exists in a dualistic relationship with the human actions which create, enforce and transform it. This definition aligns with Giddens' conception of structure. In Giddens' own terms structure is "...the rules and resources recursively drawn upon and reconstituted in processes of interaction" (Giddens 1989: p253).

...structure is what gives *form* and *shape* to social life, but is not in *itself* that form and shape - nor should 'give' be understood in an active sense here, because structures only exist in and through the activities of human agents (Giddens 1989: p 256).

For Giddens, structure both enables and constrains human actions, though not exhaustively due to the unintended consequences and unacknowledged conditions of actions.

As structure is a product of human interactions, and human interactions are highly variable, structures are correspondingly changeable as they are continually recreated. Reflections of formal structures, such as hierarchies, models of work processes or assessments of organisational culture are thus only manifestations of structures, interpreted at a given point in time. Yet human interactions are largely shaped by the structures in which they take place, and so there is some "sedimentation of institutional forms" (Giddens 1979). Giddens calls these sedimentations "systems", which he defines as "reproduced relations between actors and collectivities, organised as regular social practices" (p 66).

Contextual issues are interrelated with the actions of the systems developers and users as they create the new system. These issues provide meaning and boundaries to their actions, in a sense providing a structure in which their actions could take place. At the same time, these contextual, or structural issues are partly reshaped via the actions of developing the new system. Thus the process of systems development can essentially be described as a process of structuration. Stated another way, the context of an organisation forms the structure for defining people's actions as systems developers have to create a system which is deeply embedded in their organisation. Yet at the same time, the actions of those creating the system helped change these contextual elements. This illustrates that the structure of organisations is created via the actions of those involved while these actions are largely shaped by the context, or structures in which they occur. Hence, information systems development provides a very good example of the duality of structure and action (Orlikowski and Robey 1991).

Hence the context of information systems development can be likened to the institutional properties which structure people's actions. The difference between context and structure is that the former impacts on the systems development process while structure shapes the actions of individuals. However, since systems development consists of human actions and the context of the process largely structures these actions, there is very little difference between the two concepts in practice, so illustrating the difficulty of using the concept of structure as an element of the organisational context.

Yet is the conception of process equivalent to Giddens' conception of action as Walsham argues? For Walsham, process refers to social interactions occurring over time. Giddens argued there are three forms of temporality:

- the *durée* of activity, or the ongoing flow of everyday life;
- the *dasien*, or the lifecycle of the organism; and
- the *longue durée*, or institutional time and the development of social institutions (Giddens 1979).

Walsham's conception of processes over time focuses on the first form of temporality particularly well by illustrating the importance of social interactions, a point often ignored by the literature on systems development. While systems development can be usefully viewed as a process of social interaction, Giddens' identification of the three forms of temporality suggests *only* focusing on this micro-level is not a full analysis of the process. As Pettigrew (1995) suggested, a contextualist analysis should include multiple levels of analysis. The same can be said of a process model – it should

include multiple levels of analysis, such as the three levels of analysis suggested by Giddens.

This study focuses not only on what Giddens' terms the *durée* of micro-level everyday social activity, but the *dasien* and *longue durée* or macro processes of change in organisations. In less theoretical terms, this is an analysis of the relationship between:

- the micro-social processes through which such macro processes are enacted on a day-to-day level, such as passive and active user involvement, the negotiation of meaning and interest, the creation and sustainment of coalitions of meaning and interest and the application of expertise and authority (see Part 3);
- macro models of process used in systems development, such as the systems development lifecycle model, rapid applications development and so forth (see Part 4); and
- the institutional level of change in organisations, as social structures are created and recreated (see Chapter 4.5).

To recapitulate, many normative texts are criticised for not considering contextual factors. However, writers such as Kling (1992) have illustrated the importance of such issues and a consideration of the context models available suggests that how we consider the context of systems development will significantly impact on how we view the process. Due to the emergent nature of the organisational context, there is an ongoing relationship of influence between the context, process and content of change. A comparison between Pettigrew and Walsham's concepts of context and process and Giddens' concepts of structure and action suggests, while context and structure are broadly equivalent, the concepts of process and action are only partially so. Thus, while Walsham's process model usefully identifies the social interactions which make up systems development, it does not consider the broader processes of development operating according to Giddens' conceptions of *dasien* and *longue durée*. These different levels and the relationships between them need to be examined if we are to gain an adequate insight into the process of systems development in organisations. This is achieved in the remainder of the dissertation.

3. CONTENT

– Establishing the outcomes of the change process

Part 2 introduced the organisational context of the LSP. Particularly, it illustrated that the systems developers and users had significantly different frames of reference. This had a huge impact on how the process unfolded and the content of the resulting changes. Part 3 focuses on the micro social processes which established the outcomes of the LSP change process. It will illustrate that it can be primarily described as the negotiation of meaning and interest. This stands in contrast to the majority of literature on systems development, which does not generally recognise the political nature of change.

Chapter 3.1 introduces the dominant model from the normative literature and illustrates that in the initial stages the actions of those involved in the LSP was largely aligned with it. However, with the response of key user groups, it became clear the realisation of these plans would be problematic. Chapter 3.2 focuses on user involvement and differentiates between active and passive involvement. Implicit in much of the normative literature is the assumption that involvement will lead to commitment when users know what is happening but this is questioned with an examination of the often very active role users can play. Active involvement underpins effective negotiation.

Chapter 3.3 then argues this socially dynamic view of the systems development process strongly suggests the process can be described as the *negotiation of meaning and interest*. One key area of negotiation is *obtaining commitment* to the project, as there needs to be a coalition of meaning and interest before and during a change process. This is discussed in Chapter 3.4 with reference to incidents from the LSP. Negotiation is a very useful concept, but it does not explain the often one-way flow of information and meaning which occurred during the LSP. These one-way flows can generally be described as the *application of expertise* and the *use of authority*, as is introduced and illustrated using examples from the LSP in Chapter 3.5. Viewing systems development as a process of negotiation reflects a political metaphor of an organisation and rests on the assumption that organisations are political entities consisting of people with different worldviews and interests. A risk with adopting a political stance is that one only focuses on conflict as people act in accordance with their separate worldviews. Social process themes, such as negotiation and the application of authority and expertise, focus on the resolution of conflict that is necessary for effective systems development and implementation.

Finally, Chapter 3.6 outlines what changes have resulted to date due to the LSP and the associated negotiation of meaning and interest. This analysis will reveal that only some of these changes were anticipated and some of them have emerged over time.

3.1 Systems development planning as a political process

Although generally described in rationalistic terms, the process of planning and defining a systems development project is, in reality, overtly political. Here the political nature of systems development project planning and definition is analysed through an examination of normative literature and the early stages of the LSP. This analysis sets the scene for the following chapters, which further investigate the socially constructed nature of information systems and the social processes by which they are achieved.

The focus of this chapter is on the process by which information systems projects are defined and planned. According to the normative literature in the area, the content of such projects is intended to be aligned with broader organisational strategic plans and goals. This is generally done through the creation of a business case document which is then used as a guide for future actions. The following functional requirements document will define actions for achieving the project's goals, as established in the business case document. These norms are closely aligned with the rationalistic managerial assumptions and an alternative approach to planning is outlined while implications for systems development practice are discussed.

3.1.1 A systems development project

A project is a bounded area of desired change, its success judged according to how well its actual outcomes match the desired ones. McLeod and Smith (1996) define a project as "...a coordinated effort, using a combination of human, technical, administrative and financial resources to achieve a specific goal within a fixed time period" (p 1).

Here, a systems development project is defined as any project which involves the implementation of a new system in its organisational context. This may include the development of a new technological system, or the implementation of pre-existing or modified packages. For many (eg McLeod and Smith 1996) "systems development" includes only the creation of a custom-built technological system from scratch, but this focuses only on the technology and ignores the fact that systems also include processes, people and so forth. Here, systems development projects include the implementation of a pre-developed application, with or without modifications and the development of systems by the people who will use the system (end users). It would perhaps be less misleading to use the term "systems implementation" but there would still be confusion because, for many, this term is merely a single stage in the whole process of creating a system.

3.1.2 Defining the scope and purpose of a project

By definition, any failure in implementation is a failure in planning (Majone and Wildavsky 1978). Projects must be carefully defined in terms of the purpose of the organisation and the requirements of those who can affect and be affected by the project and the system resulting from it. The scope of the proposed project must also be carefully considered.

In an ideal world, the goals and purpose of a project are defined via a strategic planning process. In practice, of course, these strategic plans can also be retrospectively fitted to planned projects (Mintzberg 1993). In any case, Galliers, Merali and Spearing (1994 in Ward and Griffiths 1996: p 97) reported that "improving IS strategic planning" was seen as one of the most critical issues by IS and non-IS managers in a survey conducted two years before. This is consistent with previous surveys and they anticipated that the trend will continue for some time.

Most literature in this area emphasises the importance of planning. For example, Truex (1993) commented, that "The IS literature generally takes the position that failure to plan is equivalent to planning to fail", with a lack of planning leading to data integrity, inconsistency and incompatibility problems. Normative texts generally describe a top-down, pre-planned process, emphasising the need to develop a clear plan at the beginning of the development process which carefully ties the project to the broader objectives and strategies of the organisation and defines the content of the project at the beginning of the process (eg Eliason 1990; Shelly, Cashman et al. 1995).

Truex suggests the IS literature generally adopts a three stage model of information systems planning. Firstly, there needs to be a comprehensive organisational plan. This is followed by an information systems plan outlining a portfolio of organisational information systems which is derived from the first stage. Finally, the specific applications and systems are defined, developed and implemented and are justified with references to the previous two stages of planning. As Truex points out, this approach assumes a relatively stable organisational context. However, if we accept that information systems planning and development occur in a changing organisational context (or what Truex would term an "emergent organisation"), there are significant implications for how we view the process of planning, developing and implementing an information system.

Ward and Griffiths (1996) provide an in-depth discussion of this area and a concise and comprehensive literature review and evaluation. Their detailed analysis provides some depth to an area characterised by over-simplistic prescriptions. Ward and

Griffiths recognise that planning occurs in a rapidly changing context, to which it must respond, as is illustrated diagrammatically in Figure 3.1.1.

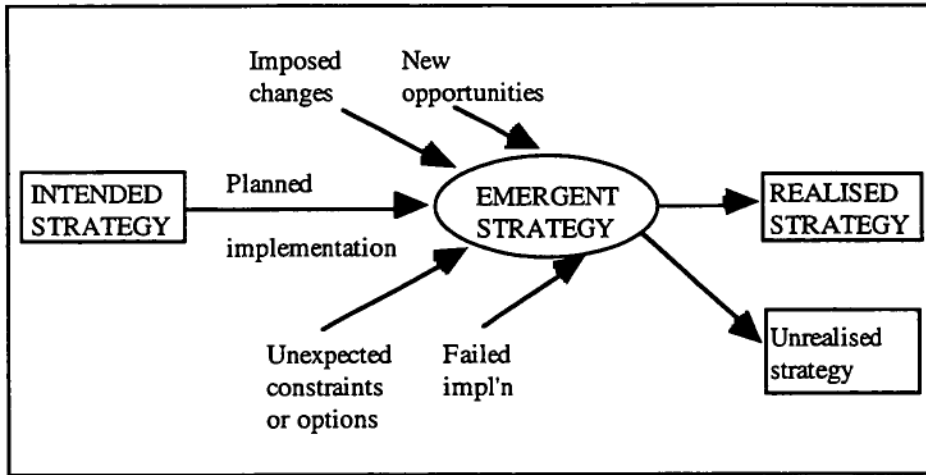


Figure 3.1.1: The realities of strategy development, after Johnson and Scholes (1993) (source: Ward and Griffith 1996, p 64)

However, despite its depth and overall excellent consideration of this area, Ward and Griffith's discussion still ignores the socially dynamic nature of organisations and the decision-making process⁹. They comment that, despite a belief in its importance, there have been some problems with strategic planning. They report many plans are not implemented because there is not enough *commitment* to them (p 97). These plans...

...were derived from a thorough *investigation* of business needs and priorities, *driven* from business strategy and objectives, and *constructed* by business teams. They may have got as far as to *obtain* the sought-after sign-off from the board, but were then left with the IT department to find a way to get them implemented, while the business got on with its "real" job of running the business (Ward and Griffiths 1996: p 98; italics added).

Top management commitment is especially important but often difficult to obtain (p 98). A survey investigating barriers to effective strategic planning found problems included *political problems*, *user-education* resources, *doubts* about benefits and middle and senior management *attitudes* (Wilson 1989 in Ward and Griffiths 1996)¹⁰. Another survey of UK companies reported that the business consequences of not having an IS/IT strategy included:

- a loss of control of IS/IT investments, leading to individuals or departments often striving to achieve incompatible objectives through IS/IT,
- problems caused by IS/IT investments can become a source of conflict between parts of the organisation,
- localised justification of investments can produce benefits that are actually counterproductive in the overall business context,

⁹ As discussed in Chapter 4.3, they recognise chronological emergence but not hierarchical emergence.

¹⁰ These represent 5/11 barriers identified by Wilson (1989 in Ward and Griffiths 1996: p 99)

- systems on average have a shorter than expected business life and require overall considerably greater IS/IT spend (sic) to redevelop more frequently than should be necessary (Ward and Griffiths 1996: p 99).

All but the last of these problems can be summarised as *conflict* resulting from differing perspectives and a lack of consensus building. These explanations suggest that politics is a real problem for the strategic planning process yet it is rarely mentioned in the normative literature on how strategic planning should occur. Look at the words in italics describing the way plans were developed: *derived; investigation; driven; constructed; obtain*. Such words and phrases strongly reflect objectivist assumptions which contrast sharply with the recognised reasons for problems: *commitment; education; attitudes; doubts; political conflicts*. These words all reflect subjectivist assumptions including the recognition of differing viewpoints, the lack of consensus between them, and the need to resolve them. This trend is reflected in phrases throughout the systems development literature generally. For example, Lederer and Sethi (1992) suggest that implementing strategic plans is harder than creating them.

Politics and pluralism are thus recognised as a source of problems but not as attributes of the process when conducted successfully. While the normative literature defines reality as an external entity to be observed and analysed, the explanations for many of the problems encountered can be attributable to issues of differing perspectives, meanings and interests. The objectivist assumptions of the normative literature do not explain the problems which occur, suggesting that these objectivist assumptions inadequately describe the process when it unfolds successfully. The argument here is that the process unfolds successfully because political problems are resolved and fails when they are not. Success as well as failure is attributable to politics in strategic planning and systems development generally.

3.1.3 Systems development as a political process

The political nature of the systems development process has long been recognised (Keen 1981; Markus 1983; Walsham 1993). In his seminal paper on computers and organisational change, Keen (1981) referred to two types of rationality that could be adopted: economic rationality and political rationality. Economic rationality strongly reflects rationalistic assumptions whereby differing viewpoints are considered irrational and conflict unnatural. Political rationality accepts that people have different viewpoints, "depending on where they stand" and so conflicts of interest are inevitable. In the former stance, different viewpoints are reduced to "resistance", implying a degree of neo-Ludditism, while Keen, in adopting political rationalism, redefines resistance as "counter-implementation". The term, "counter-implementation" implies the existence of different points-of-views and sets of

interests rather than abnormal or irrational opposition. Walsham (1993) pursued this further by employing a political metaphor of organisations, as described in Chapter 2.2.

Surprisingly, given the recognised impact of political processes by writers such as Keen (1981) and Walsham (1993), most recent normative literature still clings to (economic) rationalistic assumptions. At the most, such literature recognises that IS planning is a "learning process". As IS and business people learn more about the others' area of expertise, IS strategies need to be "marketed" or sold "to ensure that optimal support and co-operation are obtained from the organisation", and communication is important (for example Ward and Griffith 1993: pp 131, 108, 133). The political nature of the process is thus strongly implied yet rarely openly examined, reflecting the strong technocratic tendencies of the field, so that "Conflict is viewed as an irrational vestige that must be overcome in the interest of productivity and technical rationality" (Burris 1993: p 160).

The systems development planning literature is broadly aligned with the general management literature on planning and reflects the same rationalistic assumptions, yet alternatives in the general literature have been identified. Mintzberg (1993) examined planning in depth and suggested three fallacies of formal strategic planning:

- the idea that actions could be predetermined;
- the belief that planners can be detached; and
- formalisation at the expense of intuition and creativity.

He suggested that formal strategic planning methods had some real pitfalls, including an "objective" detachment undermining commitment and inducing conflict, a tendency towards conservatism and an obsession with control leading to inflexibility and conformity. Quoting Wildavsky (1973), he suggested that people tend to not attribute problems to the planning process itself.

[planners] ...are confirmed in their beliefs no matter what happens. Planning is good if it succeeds and society is bad if it fails" (Wildavsky 1973 in Mintzberg 1994: p 137).

Wildavsky (1979 in Mintzberg 1994: p 189) also suggested that planning is sometimes used simply because it symbolises rationality, with key words, such as "systematic", "efficient", "coordinated", "consistent" and "rational", being used repeatedly.

3.1.4 Planning and defining an information systems development project

Much of the normative literature on systems development follows the same top-down, rationalistic assumptions which Mintzberg criticised in relation to strategic planning.

This is not surprising, since they are interconnected (Burch 1992; Frenzel 1996). Systems development projects are to contribute to the strategic, or business goals of the organisation, and the very early stages of such projects are directed towards achieving this alignment. The resulting business case document basically outlines investment in resources required and the returns on this investment, including both tangible and intangible costs and benefits in terms of organisational strategies (Frenzel 1996, Smyrk 1997).

Referring to work by Quinn, Mintzberg and Waters amongst others, Walsham (1993) alternatively focuses on strategy formation as a process of continuing discourse. This discourse is described as a way of communicating meaning, norms and values and is linked to power relationships, hence the role of language is central. Walsham comments that communication in the discourse of strategy formation "... is inextricably interlinked to the maintenance and change of power relations between the parties involved in the acts of communication" (p 158). Thus alternative views have been expressed but are not widely diffused.

If the systems development initiative is approved, a *Business Plan* document will outline how the project is to be conducted (ISU 1997). The CIPU defined a Business Plan as a high-level management document owned, maintained and utilised by a project's steering committee to ensure the delivery of defined project outcomes (LSP Business Plan v O.A). Additionally, systems analysis will generally be conducted to detail the perceived problems and document possible solutions in what is generally termed a *functional requirements* document. Thus the process of defining a project begins with very broad goals and moves towards more specific and detailed ones. The functional requirements document is "signed off" by the proposed users of the system and the client organisation and is used as a framework for future actions. This document "...serves as the baseline document from which systems design will begin and against which the operational system will be measured in terms of its performance, accuracy, and completeness" (Shelly, Cashman et al. 1995: p 5.15). Shelley et al liken it to a contract between systems developers and users. Thus defining the requirements is a crucial stage of systems development. Systems requirements are defined by analysing the existing system and proposing a new system based on the requirements outlined in the functional requirements document. These systems analysis and design activities are considered crucially important, and so tend to be a focus of the systems development literature.

Systems analysts aim to understand the existing system as a basis for suggesting changes to it, and the objectivist nature of the literature is reflected in the language used to describe this activity. (Eliason 1990) defines it as, "...the systematic *collection of facts and opinions* relating to how the current system works and the new

system requirements" (p 128, *italics added*). (Martin 1995) calls it "problem *detection*", where user expectations are *gathered* and the information systems department *generates* possible solutions which it presents to top management. Analysts *capture* information and *diagnose* the current situation, which is seen to be an objective fact and *identify* problems (for example, Eliason 1990; Jordan and Machesky 1990). Political issues or issues of conflict are also noted as objective elements which "*require identification and evaluation*" or *synthesis* (Eliason 1990; Jordan and Machesky 1990. Shelly, Cashman et al's. (1991) description of these activities is quite straight-forward:

- (1) Get the facts;
- (2) Analyse the facts;
- (3) Make a decision (p 3.4).

The activities are likened to a doctor examining a patient, a police investigation of criminal activities, solving a murder mystery or completing a jigsaw puzzle (Martin 1995; Hoffer, George et al. 1996). The organisational situation under examination is an objective fact for the analyst. Only a few normative texts question this general stance. In a survey of commonly used normative texts on systems analysis and design (see Appendix 2), more than half of them reflected what Davis (1992) termed "a simple, rational expert-driven model" (p 14), where only two roles were acknowledged: a passive user and an active analyst. As Davis (1992) pointed out, "A theory of requirements determination is not only one of elicitation; it is also a theory of communication between analyst and user" (p 15). As is discussed in the next chapter, the problem with this common approach is that it relegates users to passive participants.

The rationalistic stance of this normative literature has been criticised for its inability to describe what actually occurs during the process of systems analysis. Westrup (1996) is critical of many existing techniques of requirements analysis as they are based on limited organisational theory. He argues that attempts to improve them have been problematic for three reasons:

- they tend to employ very abstract and simplified theories and downplay critiques of them;
- they tend to move unquestioningly from descriptive to prescriptive theories; and
- they tend to ignore the practices of systems developers which may embody sophisticated organisational awareness which is not expressly in their techniques.

Reflecting on structured systems analysis and design techniques, he commented:

In short, these techniques conjure up a tidy world of functionality, harmony and clarity in which Ockham's razor holds sway; one that is removed from the muddle, conflict and short-termism so common in systems development (Curtis Krasnoe and Ascoe 1988) let alone organisations in

general. This helps explain its attractiveness to management and systems developers and also its prime failing (p 161).

Thus while the bulk of the literature on defining information systems projects adopts a rationalistic approach, alternative perspectives are available. In general, though, the planning literature on systems development is overtly objectivist and rationalistic and does not recognise the political nature of change. As argued in the remainder of the dissertation, this is a major oversight.

3.1.5 Defining the LSP project

The commercial system development methodology employed during the early stages of the LSP reflected the normative literature to a large degree. The commercial systems development methodology employed assumed a rationalistic stance in common with most of the normative literature in this area, but did allow for active involvement by other parties at some times to review the process. Those involved in the project used the methodology selectively as they saw appropriate, but it also greatly influenced their actions. Strategic planning in the methodology involved the *collection* and *analysis* of information, the *determination* of IT potential and a cost benefit analysis, with a final stage for *developing* and *negotiating* the final plan.

Thus the normative literature the systems developers formally employed is broadly aligned with the dominant rationalistic approach. Here the focus is on whether their actions did as well. Subsection (a) outlines the impetus for the project while the creation of a business plan is discussed in subsection (b). These incidents suggest that the actions of those involved with the LSP broadly did align with the dominant objectivist model at this stage. However, the drafters' reaction to the prototype described in subsection (c) suggests problems were looming with this approach, while the creation of the functional requirements document is outlined in subsection (d).

a) Impetus for the project

The broad justifications for the project were introduced in Chapter 2. 1, but the actual impetus for the project seems to have come from two interconnected sources: ongoing printing problems and the active promotion of reform in this area by an executive manager.

Carefully proof-read material leaving the OPC would sometimes be printed with sections missing or in differing formats. The chief drafter discussed these issues with the executive manager to whom he was responsible, who foresaw a technological solution. An analyst from the CIPU, who was asked to investigate the issue by the executive manager, observed a number of problematic or inefficient issues and

highlighted procedural inadequacies beyond the scope of the perceived problem. The problem area became redefined, promoting the need for a project such as the LSP.

The executive manager strongly believed that the process of producing legislation could be greatly improved in terms of its efficiency and effectiveness if computerised technologies could be used as aids. The manager was aware of complaints concerning the process of producing or amending legislation and of problems resulting from the inadequate state of the statute books. He foresaw a technological solution, called for a review of the processes for producing legislation and strongly promoted reform in this area. The Department's Secretary reported to the Premier,

The Office of Parliamentary Counsel is experiencing serious difficulty in keeping pace with the volume of drafting work generated by instructing Agencies and Members of Parliament. As a result, important parts of the Government's legislative program are facing unacceptable delays.

This is an important machinery of Government issue for which there is no simple solution and a range of measures is being implemented to tackle it. The Office will soon seek to recruit an additional experienced Parliamentary Counsel but this may prove impossible as such people are in short supply throughout Australia.

A second measure is a review of the process used to produce legislation which will be carried out by the Corporate Information Projects Unit of this Department. The review team's principle task will be to identify opportunities for streamlining the process. This will allow the Office's resources to be concentrated on the highly skilled aspects of drafting by automating or eliminating more routine tasks where possible (Briefing Note from Secretary of DPAC to Premier 22 June 1993).

At the same time as the OPC were reporting problems, the related Legislation and Policy Division were preparing a draft budget application for consolidating the Tasmanian Statute Book (letter from legal officer to executive manager 28/6/97). The proposed review of the legislation production process was extended to include the consolidation of the statute book.

At this stage, broader economic, efficiency and social impacts of system, as defined in the later versions of the Business Case Document and Chapter 2.1 of this dissertation, had not been concisely identified. The project essentially needed an immediate concrete impetus such as identifiable printing problems. Again, problems with printing would not have led to the LSP without the involvement and commitment of the executive manager.

Initially, the project was presented in documents as simply a review of existing processes, the preliminary brief outlining the purpose of the review as:

... a systems analysis of the legislation production process and the process of consolidating and maintaining existing legislation in an accessible format. The review does not include parts of the legislation production process which involve parliamentary procedure except when relating to OPC (30/6/1993).

There was no mention of any outcomes of the analysis at this stage and the "client" was the Secretary of the Department. However, the executive manager was committed to reform and a draft business case document was produced the following month. In an email to the project manager, the executive manager wrote,

I would rather kill before it started any thought that the project and need for reform was negotiable. The question must be how we proceed forward (8/5/93).

While conditions justifying the LSP had been present for some time, the printing problems and vision for reform by the executive manager were a trigger for initiating the project, but were not part of a formal strategic planning process. This comment by the executive manager hints at the political nature of the change process, but this is not reflected in the methodology employed by the systems developers nor the systems development literature generally.

b) Creation of a business case document

The business case document outlined the reasons and objectives of the project and was produced iteratively through discussion sessions with interested parties by the CIPU. In August 1993, a draft was produced as a discussion paper for key people involved in the processes of producing legislation, including members of the OPC, the Attorney General, the government printer, parliamentary liaison officers and the executive manager promoting the project. The aim of these meetings was to discuss and review the project as a whole and to build commitment to it. Several meetings were held over the next months, with some members of the consulting group eventually becoming the steering committee.

There were in all, two draft versions of the business case and at least eight later versions. Comparisons between different versions of the document illustrate:

- a growing understanding and better articulation of the problem area;
- outcomes of negotiations and social interactions surrounding the process;
- the effect of changing circumstances; and
- attempts to improve the implementability of the proposed system by changing the scope of the project.

i) better understanding and articulation

Over time, the business case document naturally became more concise and detailed, reflecting the amount of effort spent on it and the creators' growing understanding and articulation of the problem area and possible solutions. As well as including much more detail, comparisons of the business case document illustrate how definitions of the objectives of the project were revised and became more concise and specific. For example, the first draft of the business case did not clearly include the concept of automatic consolidation, aiming only for "improved access to legislation by issuing regular consolidations in an easily accessible format". The project at this stage aimed to improve the efficiency of drafting processes but did not specify how. Later

versions of the document specifically mentioned computer-aided drafting facilities and automatic consolidation. Project strategies were also not documented until later versions, indicating the process by which the process unfolded was at least partly emergent. This issue is pursued in Part 4.

ii) social interaction

Changes in the different versions of the document were also a result of discussion and compromise between the different parties involved. The initial planning sessions discussed issues such as the objectives, scope and cost/benefit analysis of the project, project risk, risk management strategies and development strategies. These discussions resulted in many changes to the document, suggesting the social interactions in these meetings were significant. For example, while in the draft version, one of the aims had been to "eliminate inefficient work practices (eg. hand-written drafts, manual text searching and retrieval of existing legislation)", later versions (v 1.1 onwards) amended the "eliminate" to "minimise". The second planning meeting included a demonstration of some technology which may have been useful to illustrate some concepts, suggesting some involved were not totally convinced. The second draft version of the business case included large sections justifying each aspect of the document and explaining the actions of the systems developers, suggesting a perceived need to "sell" their actions and the document itself.

Those promoting the project also needed to "sell" the project as a whole and this is also reflected in the business case document. In producing a submission for Cabinet and the Budget Committee, the CIPU and others involved in the project created a detailed justification for the project and this nine-page document was included in subsequent versions of the business case. At this stage most of those involved in the project were probably aware why the project was being instigated, but these reasons had not been formally articulated. The inclusion of such information indicates not only the amount of effort spent on the document, but also the desire to sell the project to others. It hints at the negotiation involved in the planning process.

iii) changing circumstances

The differences in project risks perceived in different versions of the business case gives an indication of the effect of changing circumstances and of the concerns of the systems developers at different stages in the project. They were periodically revised "in accordance with the current stage of the project" (4/8/95) as some issues were resolved and others emerged. When the business case document was reviewed just prior to steering committee meetings, new risks were identified or the status of existing risks was reassessed.

Interestingly, the OPC and the issue of user acceptance were not immediately identified, though non-acceptance by parliamentarians was. Later drafts developed during the planning sessions documented concerns that the OPC (now termed the "client" rather than the "user") did not place a high priority on the project and that client participation and knowledge about the technology and application were an issue. Later concern about the OPC's knowledge about the technology was reduced, following training sessions and greater exposure to computerised technology.

The OPC's involvement was an ongoing issue throughout the project. To the CIPU, the OPC often seemed unwilling or unable to effectively contribute to the project. Early in the project the systems developers expressed their concern that the OPC did not place a high priority on the project while later they emphasised the "business owners' capacity to participate". This perhaps reflects a growing awareness of the pressures under which the OPC operated, though, upon reading a draft of this analysis, one system developer commented that she saw the two issues as distinct, with the latter emerging some time after the former. At the same time, on reading early drafts of this dissertation, several drafters commented they had been willing to be involved, but were not given adequate opportunities.

Conceptions of organisational change issues also changed over the course of the project, and this is also reflected in the risk assessments. Early in the project, a need for physical restructuring in the client area was identified. After the planning sessions and the first prototype, this risk was no longer mentioned, but the impact of the application on the client's organisational structure was. Around the time the functional requirements document was being assessed by the OPC, this risk seems to have been replaced by a concern that procedural changes imposed by the system could be a problem, and this concern remained for quite some time.

Such risks also reflect trends indicated in other sections. The systems developers' assessment of risks echoes other observations that in the early stages of the project they tended to underestimate the involvement and influence of the OPC, for example. This reflects the emergent nature of the overall change process, and it stands in contrast to most of the literature on systems development.

iv) changing scope

Differences between versions of the business case also reflected a need to change plans in order to make them implementable. When the project's scope became problematic, it was narrowed slightly so that public access to the system was defined as a separate project and the history of some acts was not included. Later versions of the business case carefully defined areas which were not part of the scope of the project. For example, it indicated the project would only produce drafting facilities

for members of the OPC and would not change the way the OPC's clients provided instructions. In December 1995, the scope of the project was amended by,

... deleting the unresolved objectives and including them in the list of objectives that are not part of the project (LSP Progress Report to Steering Committee, 13 December 1995)

Thus while the LSP initially included providing access to automatically consolidated legislation, later this was defined as a linked but separate Legislation Publication Project (LPP). Thus, the project's objectives changed slightly over the course of the project. This suggests the emergent nature of the process, as will be explored in Part 4.

Version 1.1 of the business case was presented to the first meeting of the steering committee brought together to oversee the LSP. The committee included the chief drafter, the executive manager responsible for the OPC and CIPU and other senior public servants whose organisations had a stake in the project. Several politicians were also included and the leader of the CIPU also attended. This group was essentially the "top managers" leading the project. The membership of the committee was derived from those who attended the first planning sessions for the project and nine meetings were held during the course of the project. More frequent meetings were impossible given the seniority of those involved and meetings were timed to coincide with key milestones in the project, such as the completion of the functional requirements, the completion of the tendering process and the initiation of system implementation.

Most of the time the committee played the role of a review body to check the progress of the project but occasionally they had an important decision making role. Many items of the agenda were simply items to be noted or approval granted. For example, the committee was asked to approve some changes to the business case and other documents and the choice of tendering company. Even when the project was greatly delayed, the steering committee merely noted the slippage and the measures made in response to them. Occasionally, though, the steering committee or individual members on it made decisions directly in response to changing circumstances or more information. For example, one member of the steering committee resolved a long-standing issue of paper-size. Legislation had long been printed in B5 sized paper and, with A4 becoming the norm for printers and photocopiers, the CIPU and OPC spent a considerable amount of time evaluating the costs and benefits associated with a change to A4 sized legislation. They concluded there were few cost-savings to be made and there was also the risk that members of the public and parliamentarians would not appreciate the larger volumes and the OPC could sidestep the issue by printing B4 format on A4 paper which could be cut down to size. Following a review of this decision by the steering committee the Attorney General decided to

change all legislation to A4 size. The project team had to comply with this decision, illustrating how decisions made higher in the organisational hierarchy of government helped shape the actions of others. This influence of hierarchical authority indicates the political nature of change, and is pursued further in Chapter 3.5. In essence, changes to versions of the business case suggest the planning process was far more emergent over time and involved much more negotiation than implied by the literature in this area.

c) The first prototype

To aid definition of the project and help members of the OPC visualise what computerised technology was capable of doing, the LSP team organised for external consultants to produce a prototype in early 1994. The aim of the prototype was to help define functional requirements, involve the OPC in the LSP to "get OPC staff use [sic] to the concepts" (Consultants' working notes, no date provided) and evaluate what seemed to be some promising software for a computerised drafting environment. The prototype was created with only limited input by members of the OPC until being trialed, despite moves by the LSP team to involve some of them. The exercise was later judged to be unsuccessful by the CIPU but interestingly, this is not reflected in the review forms created at the time. Nor do the reasons for why the exercise was considered unsuccessful align with the formal objectives of the exercise, suggesting there were some other expectations. While the prototype produced can be judged unsuccessful, the process of producing it provided a valuable learning exercise for those involved. This incident illustrates that systems development can and should be viewed as learning and negotiation.

Later the drafters generally judged the prototype negatively. One drafter later said she had been horrified while another thought it had been "unfriendly" because it was not intuitive to use. Another drafter later wrote:

I was away when the prototype... was given a trial in the office, but from all accounts it was quite inappropriate (transcripts 5/7/94).

The company who had produced the prototype commented afterwards that there was a lack of involvement from OPC as they only had the chance to review it and had not participated in its creation, stating:

While the prototype system seems to have had some positive results in assisting with the analysis of requirements, the OPC (user) perspective is that the Prototype was not successful (Consultant's working notes, no date recorded).

The negative reaction by the drafters could be partly attributed to the short amount of time they had to assess the prototype and also their lack of experience in computerised technology. The drafters were only given one to three hours each to formally assess the prototype and while the technology was in the OPC office for

several months, several people commented they were allowed very little time to use it. For some, this was one of the first times they had extensively interacted with computing technology. The systems developers realised this and provided explanatory notes for the drafters about pull-down menus, toolbars, navigation in and between screens, help facilities and so forth. In effect, the review for the drafters would have been more a course in applied computer basics than an effective chance to review a particular application. Hence the drafters complained that clicking the mouse was annoying and intrusive and dismissed the software. The drafters also found it difficult to relate to the concept of a prototype and rejected the whole system once they realised it could not fulfil their detailed requirements. Importantly, the systems developers had underestimated the fact that the drafters had very little previous experience with computers at this stage. As a result of this prototyping incident, the systems developers changed the way they approached user relations later, as discussed below.

The negative reaction was also an indication to the systems developers that the technology was not appropriate. The interface used was not intuitive for the drafters and the structures used did not align with the way that the drafters created amendment legislation. The prototype allowed drafters to only view one or so lines of the text they were creating and some of them dismissed the prototype immediately as it was obvious to them they could not write using such a tool. One drafter commented that the short time they had been given to assess the prototype was irrelevant, as they could immediately see it was inadequate.

Yet this contrasts with the review forms filled in at the time. The issues raised on the review sheets were generally of a minor or detailed nature and, written by the then new business analyst, reflected her limited understanding of the drafter's world at that time. For example, principal legislation is referred to as "primary legislation" in these documents. Issues raised included terminology used, document sharing, draft naming conventions and so forth. There is no indication of the drafters' overall impression of the prototype. No major problems or overtly negative responses are reported. Upon reading an early draft of this description, the business analyst commented that most of the negative feedback was verbal and beyond the scope of the review form.

The systems developers became aware that the prototype was inadequate, yet the review tools they used did not allow for such a reaction. The review forms were explained thus:

This form is intended to document any problems that may occur with the prototype system while it is being used by the OPC and also any changes that are suggested while the system is being used. For example, changes may consist of requests to add functions, modify or alter screen displays, delete or remove information fields, etc.

There was no option in the review form for abandoning the whole prototype.

The LSP team also judged the prototyping exercise as unsuccessful due to three reasons, as is indicated in Table 3.1.1. They were also later concerned that the prototype gave a bad impression of their work.

Objectives of the prototyping exercise	Reasons given for why the prototyping exercise was considered unsuccessful
<ul style="list-style-type: none"> • to assist with the definition of functional requirements • to involve the OPC in the LSP, ie. developing a sense of ownership within the client area and a natural commitment to the project; and • an evaluation of the specific software package employed (Draft Functional Requirements Specification 15.1) 	<ul style="list-style-type: none"> • the prototype system did not satisfy some of the clients' basic drafting requirements which prohibited their full evaluation of the system; • the clients had trouble using the system software; and • the clients didn't relate well to the concept of identifying key business functions (draft functional requirements specification 15.6-7)

Table 3.1.1: Documented aims and evaluation of the prototyping exercise

As is illustrated in Table 3.1.1, the reasons given for why the prototyping was not a success do not match with the stated objectives of the exercise. Essentially, the objectives of the exercise tend to focus on the *process* of producing the prototype while the evaluation focuses on the *content* of the prototype itself. The output, or content, certainly was not a success, but was the process of producing a prototype a failure? The purpose of a prototype is to give people an impression of how something will look and to aid communication. The prototype did help define the functional requirements by illustrating what the users did not want and by providing a forum for discussing such issues. It helped involve the OPC in the project by illustrating to them their contribution was necessary if the resulting system was to be useable. It is doubtful the prototype itself helped to develop "a sense of ownership within the client area and a natural commitment to the project" but unless it was just a token gesture of user involvement, the fact they could determine the content of the project would aid their commitment as it would naturally become more closely aligned to their needs and interests. The prototype was also a good evaluation of the particular software package, illustrating that it had serious shortcomings for this application. Hence, although the prototype itself was not successful, the exercise was, if judged according to the pre-set objectives.

The reasons given for why the prototyping exercise was not considered a success suggest the systems developers had expectations which did not strictly align with the documented aims or that the aims changed over time. The systems developers had expected the prototype to help "sell" the concepts and general technologies involved and help them to identify some specific and detailed information¹¹. They had

¹¹ The consultants contracted to create the prototype described their understanding of the purpose of

assumed that once the users knew more about the technology, they would be happy to use it and did not seem to link user involvement with active user participation in the process leading to the development of a system the users felt comfortable with.

The systems developers simply did not anticipate the feedback they received and were unprepared for it. Yet this feedback proved highly constructive in the longer term. They expected the users to give feedback on aspects of the amendment drafting process but found that they had not satisfied their basic drafting requirements. They were looking for the drafters' help in identifying business functions but learnt that the drafters were unfamiliar with the general concepts employed. The LSP team had expected the prototype to be a general learning device for the users and a device for themselves to learn the specifics of the drafting process. It helped illustrate they had not understood some basic aspects of the drafting process and that the process of developing and implementing the system had to carefully consider the backgrounds, attitudes and knowledge of the users. The review noted:

The OPC drafters found the prototype system difficult to use due to their unfamiliarity with the user interface... The client's lack of experience with the use of computers... also contributed to this difficulty.

Though these factors have inhibited the use of a prototype, they have helped to identify the importance of a structured training program for the OPC...(Draft Functional Requirements Document 15.7).

Note that the systems developers still viewed the resolution of these problems as user learning. Despite this, though, as prototypes are designed to be, it was a very effective learning device for the systems developers, with one of the systems developers commenting later,

The software was "off", but we got some useful feedback (transcripts 12/7/94).

The prototype effectively illustrated the inconsistencies between the two groups' frames of reference and suggested to the systems developers that it would take more than a simple "sales" exercise to gain the commitment of the users. It indirectly helped the systems developers gain access to the OPC by illustrating the need for interaction. The project manager used the shortcomings of the first prototype to emphasise to the OPC they had to have an input into the development of the system.

It's up to you drafters to decide what you want. The last prototype, we know this is not what you want so we didn't spend any more time on it (transcripts 28/7/94).

The drafters agreed and this early prototype set the scene for their further involvement. At the same time, the LSP team were concerned that members of the

the prototype thus: "To produce a prototype that can be used by [the executive manager] to draft a current piece of legislation to demonstrate to [the chief drafter] how quick and easy it is..." (consultant working documents Prototype file, no date provided).

OPC would come to have a negative impression of the proposed technology and their abilities. As one of the external consultants commented:

The failure of the prototype from the OPC point may adversely affect the overall project (transcripts 2/5/94).

The LSP became reluctant to show the OPC undeveloped later prototypes because of the negative reactions from the first one. Nevertheless, the prototyping exercise was a useful learning experience and was instrumental in forming initial perceptions and later actions.

d) Creation of functional requirements

The experience of the first prototype had illustrated the complexity of the OPC's requirements to the systems developers and these requirements needed to be documented in the functional requirements specification document. The LSP team found that determining the precise requirements for the system was extremely difficult due to the complexity of the processes involved. This complexity was a result of several issues including:

- the individualistic nature of the drafting process. Although legislation is highly structured as a rule, writing is a creative process requiring significant intelligence and concentration. The drafters were highly trained specialists who had developed their own individual ways of producing legislation;
- the bill writing or amendment process was an iterative process between the drafters, the agency requesting it and the support staff. At any one time, for example, there may be several versions of the one bill or amendment;
- these processes were bound to the workings of parliament and government as a whole so had to be able to fit in with them effectively;
- the processes cut across several government bodies which otherwise had very little contact with each other; and
- the drafting office were responsible for the production of bills and amendments within the public service but could be completely overridden by amendments made in parliament and it was not the role of the office to track these changes.

From late November 1993 until early May the following year, a business analyst developed a functional requirements document for the LSP with the help of the deputy chief drafter. Due partly to the high workload of the office, the analyst subsequently had very little access to others in the office. A review consultant commented:

...through low participation users are not developing ownership commitment and understanding of the intended system (2/5/94)¹².

With the help of the deputy, the business analyst spent considerable time preparing the requirements document, the resulting document being just under 100 pages long. Several of the tendering consortiums later described the document as perhaps the most comprehensive they had seen. The document was also considered to be of a very high standard by an external project management consultant.

The draft functional requirements document is well written, comprehensive, and to such a level of technical detail that I now question the need for the development of a separate generic design document...

The existing functional requirements are more than adequate to approach the marketplace with a Request for Information (RFI) from potential suppliers... (Project Management Consultant's Report 20/6/1994).

However, at the same time, the consultant commented:

The reaction of the OPC in requesting more time to consider the functional requirements is understandable, and I support the decision to delay sign-off by the OPC until the staff have had time to fully evaluate the functional requirements (Project Management Consultant's Report 20/6/1994).

Obviously the consultant did not expect the OPC's evaluation of the document to greatly affect the content of the project if he recommended the functional requirements document was comprehensive and ready to present to potential suppliers while the OPC were still evaluating it. They were perceived as largely passive participants in the process, an issue pursued further in the next chapter.

The draft functional requirements had been presented to OPC management early the previous month and the project was slipping while the OPC were unwilling to complete the "sign-off" stage of the task. The systems developers' report to the steering committee on the 15 June 1994 reported,

The Chief Parliamentary Counsel expressed concern that insufficient time had been allowed in the project time to allow OPC staff to understand the Functional Requirements specifications and to evaluate the full implications of automatic consolidation. He was of the view that unless the OPC staff were given the opportunity to gain a full understanding of the Functional Requirements specifications they could become alienated from the project and reject the system as a technological solution imposed on them without proper consultation (CIPU Report to Steering Committee, LSP project 15/6/1994).

Unfortunately for the systems developers in the short term, but fortunately for the project in the longer term, though, some OPC staff were determined to be active participants in the process. This is pursued further in the following chapter.

¹² Note commitment was expected to the system (i.e. content of project), not to the project itself.

3.1.6 Reflections: A gap between theory espoused and practice

In summary, most systems development literature is strongly aligned with the rationalistic managerial assumptions outlined in Chapter 2.2, while an alternative perspective articulated by Mintzberg(1994), Walsham (1993) and Westrup (1995) outline problems with it. But why should it not reflect such rationalistic assumptions? All that has to be achieved at this stage of a project is a plan for the rest of it that is perceived to be viable and worth committing to, so this approach could usefully imbibe the whole process with a sense of technocratic rationality (Buchanan and Boddy 1992). The problems emerge when these plans are to be implemented.

Implementation is simply seen to unfold as a result of these previously established plans yet, as will be illustrated below, this is not necessarily the case. The importance of planning has been emphasised, yet in doing so, the process of realising those plans has been underestimated. Yet, often in systems development, the literature simply focuses on the importance of planning, and relegates implementation problems to problems of change management, or implementation. To reiterate Wildavsky and Mintzberg's observation, IS planning is good if it succeeds and the organisation is bad if it fails and IS planners are confirmed in their belief no matter what happens.

In the early stages of the LSP the broad goals of the project were defined and the actions of those involved in the LSP formally seemed to be following the broad prescriptions of the rationalistic normative literature as outlined in the previous chapter. Yet problems were starting to loom on the horizon. While the rationalistic approach helped to plan the project, implementing the plans seemed much more difficult.

Comments by some of the external consultants involved with the project at this stage reflect the perceptions of the systems developers generally and suggested that problems were looming, even at this stage

User attitudes will be the biggest factor in the system's success. Implementation will look like it will be the hardest part (transcripts 4/10/1993).

The business analyst believed the people who were going to use the technical system were the biggest risk in the whole project:

OPC aren't taking ownership of the problems that the project is addressing, therefore they won't accept ownership of the solutions, which introduces a high risk to the acceptance of the overall system (LSP status report 30/5/94).

If we only considered the early stages of the LSP, then we could confidently claim the process could be described in much the same way as the rationalistic normative literature. The executive manager proposed a new system to meet some organisational goals and resolve some ongoing problems. The business case and

functional requirements documents articulated these proposals in more detail. At this stage, the process could be described using the concepts employed in most of the normative literature.

But then cracks started to appear in the polished rationalistic approach. There was one "slight" problem. Many of those involved, particularly the drafters, did not like the plans being presented to them. They and their opinions had to be considered.

3.2 Active and passive user involvement

User involvement is usually seen to be linked with system development project success but the nature of this involvement is generally not investigated. Here, a distinction is made between passive and active user involvement.

User involvement in the process of systems development is generally linked positively with commitment to the project and its outcome (Ward and Griffiths 1996; Chang 1995). As the prototyping exercise illustrated, the systems developers in the LSP initially aimed to involve OPC staff in the definition of the project so they could "sell" the project to them while finding out what some of the more detailed requirements were. The drafters' overtly negative reaction to the prototype and the systems developers response in abandoning the prototype signal more active user involvement.

The focus of this chapter is on the involvement of the people who were to use the system (the users) in the systems development process. If systems developers are experts in creating systems, what role do the users play? While most of the systems development literature emphasises the need for user involvement, what is meant by this is less than clear. Here a distinction is made between passive and active user involvement. The difference between the two has some important implications for how we view the process of systems development.

The term "user" is widely used but needs some clarification. Smith (1997) defines a user as "any employee or customer of the organisation who will be directly or indirectly affected by the system" (p 33). Users can be classified in terms of the role they play within the information system. They may be:

- *end users*, or people who directly utilise the system to perform their work tasks;
- *manager users*, who, although they may not directly use the system, have a great stake in it as they are usually responsible for identifying the need for a new or revised system and for the system's initial specification;
- *customer users*, that is, people who are affected by the inputs and outputs of an information system; or
- *system users*, or the people responsible for managing the system or applications involved in it. System users include database and systems administrators, for example (Smith 1997).

In many cases a person may take on more than one of these roles. In this chapter the focus is upon the end users, as they are the most affected by the system.

3.2.1 User acceptance and system success

Users' acceptance of a system is often a major determinant of a system's success (Lyytinen 1988). If users are not happy with their system, they are unlikely to utilise it and the developers' time and the organisation's money would have been wasted. Normative texts on systems analysis and design also recognise that user satisfaction is important. Such texts refer to the need to produce an "effective and acceptable solution", for example, or the "need for constant interaction with those affected by the change to produce a workable system" (Kendall and Kendall 1988; Hawryszkiewicz 1994). Systems development as social or political activity is thus strongly implied but not explicitly examined.

Most of these texts do not adequately explain the process by which this consensus and agreement is obtained. They seem to assume that systems developers are able to extract requirements from users that can be easily transposed into a technical solution which the users, with some training, will find acceptable. This questionably assumes that:

- users know what information developers need or developers know what to ask;
- users know what they want a new system to do or would be happy with what the developers presented to them;
- users as a group agree on what they want or, if they do not, their differing world views can be interconnected logically in an internally consistent manner;
- users are able to articulate the above in a way the developers are able to fully comprehend;
- developers are able to understand and appreciate the working conditions, experience and worldviews of the users;
- information can be passed on from analysts to other systems developers without relevant information being lost through the process of communication;
- users' requirements nor the organisational context do not change between the time of analysis and implementation; and/or
- users' requirements can be logically transposed into a technically based solution which will be acceptable to the users once they have the appropriate technical training.

This approach assumes that the communication process between users and systems developers is capable of being perfect and that users' requirements are both internally consistent and able to be translated into a technically viable solution which is acceptable to them. In other words, it assumes that humans can be wholly rational creatures, that decisions can be optimal, the planning, designing and implementation of a system is a linear process, information systems development can be isolated from

its organisational context and that communication can be perfect. These assumptions have been strongly criticised (Kling 1981; Turner 1987; Reed 1991; Boland and Greenberg 1992; Jones and Walsham 1992; Westrup 1995).

The term "systems analysis" reflects the common assumption that one important step of the systems development process is the extraction of relevant information from the users in order to design and implement a technically appropriate solution that is acceptable to the users. Yet couching the process in terms such as "analysis" and the "elicitation of information" suggests it involves the extraction of information from a static situation in which the users are passive subjects of the proposed changes. In reality, social situations are dynamic and fluid and are affected by the very process of elicitation.

This reflexive nature of information elicitation has been widely recognised in social research but tends to be overlooked in discussions on systems analysis. Famous research programs such as the Hawthorne studies in the 1940s have illustrated the effect observation could have on social phenomena being observed (Aungles & Parker 1992). While initially investigating the effect of lighting levels on work productivity, the main contribution of these studies was that productivity was more affected by the observation than the amount of light. Guidebooks for social field researchers warn that social situations can be influenced by the observer. Van Maanen, for example, suggests that the success of any fieldwork "... depends inherently on the results of the unofficial study the observed undertake of the observer" (1991: p 31). Systems analysis is thus a two-way process of interaction, not the one way process of extraction suggested by the literature in the area.

There is also much evidence to suggest that, while analysis, planning and implementation can be conceptually separated, it is difficult to do so in practice. Referring to Malhotra's (1980) study of dialogues, Turner (1987) argued that problem generation and solution generation could not be achieved independently. Discussions surrounding these activities tended to be cyclical in nature, with requirements and solutions "migrating together towards a convergence" (p 101). Malhotra suggested that the fragmentary nature of the discussions reflected the discussions' importance in stimulating cognitive processes rather than just conveying information. As Turner stated,

In summary, the common wisdom about the design of information systems is that it is an ordered process, performed at the beginning of a project (in the life cycle strategy), a methodology which when adopted will produce the same results; that is, top-down, moving from general to specific; and that definition of requirements procedures design solutions. Research findings suggest the opposite. Design is ad hoc and associative, the process is individual and experientially based. The solutions produced by different designers are usually different, much of the design process and solution and problem definition are intertwined...[this is] an observation that in the lifecycle approach it may be unrealistic to expect that requirements will ever be completely articulated at the beginning of a project (Turner 1987: p 100).

It is unrealistic to assume that the solution to the system design problem can be comprehensively planned in one step. A view that the process of systems development is "an orderly progression from requirements analysis to a solution designed purely around those requirements" (Fitzgerald 1995: p 16) interferes with ongoing and necessary social processes of learning and negotiation between users and developers (Jones and Walsham 1992). Thus, if we view systems development planning, development and implementation as a process of social interaction at a micro-level, there are substantial implications for how we view the process at a more macro level. As discussed in Part 4, this more macro process is likely to be iterative, incremental and involve substantial active participation. In a word, it can be described as emergent.

While Malhotra suggested that the iterative process of analysis and design reflected the nature of cognitive processes, a further interpretation is that it describes processes of negotiation. As people interact they discuss issues from different frames of reference and interests (Orlikowski and Gash 1994), producing a common frame of reference through learning and negotiations as interests are discussed and compromised.

Hence, systems analysis and design can be considered to be a two-way, iterative and emerging process of negotiating interests and meaning rather than a one-way, non-iterative process of extracting information. While systems developers are determining user requirements, users are discovering what the proposed changes may involve for them. Questions asked by the analyst in interviews do not just extract information, but provide information to the user and initiate negotiations concerning what is acceptable to the user when changes are made. The problem with defining systems analysis and design as the extraction of information from users is that the political nature of systems development is ignored and users are relegated to a passive role of merely providing information.

If one adopts such a rationalistic approach then user non-acceptance is simply defined as "resistance". Hirschheim (1988) suggested that resistance is a word loaded with pejorative meanings, suggesting unlawful or warranted acts, and so needs to be eradicated or neutralised. It is viewed this way because "resistance" is defined by those who see the change being resisted as positive. Reflecting Keen's (1981) argument that resistance merely reflects differences of opinions, they suggest that resistance can be legitimate and even beneficial. They point out that change in itself does not necessarily lead to resistance due to innate conservatism, uncertainty or a lack of involvement, but due to problems in the system, such as a mismatch between the system and its organisational context, poor technical quality or the personal characteristics of the designers. Resistance is, in effect, a way of sidestepping the

socially dynamic nature of systems development. If we recognise that systems development is a political process, then the involvement of users becomes not just a good way of increasing the chances of success, but an integral part of the process. A political perspective recognises that users will have an active role in the systems development process.

3.2.2 User involvement in systems development

Most recent literature on systems development emphasises the need for users to be involved, particularly in the requirements determination stage. For example, Eliason (1990) comments that wise designers try and involve users at various stages in the design process and particularly in the early stages. Martin (1995) suggests that user involvement must be "real rather than decorative" and that involvement in systems analysis and design enables users to have a sense of ownership. This is also reflected by research in the area. For instance, Kaasbøll (1995) suggested that systems developers need both etic and emic knowledge. Etic knowledge is gained from outside an organisation and this needs to be combined with emic knowledge, which is obtained from inside an organisation. Effective design, he argued, required both.

Unfortunately, the common structured approaches to systems development relegate end users to passive providers of information (Bansler and Bødker 1993). For example, Smyrk (1997) suggests that users should not be involved in the broad definition stage as at this stage the focus will be on the outcomes of the initiative and it will be easy to ask the wrong questions. In fact, he suggests that the people who will operate the system are largely irrelevant. At this stage *utilisers* such as senior managers or the organisation as a whole should be involved. Users should only be involved in a project after it has been defined. They decide the colour of the screen, not the definition of the whole initiative. If users then did not use the system or used it in a way that was not envisaged, the issue was one of change management, industrial relations or perhaps some systems design problem. This model thus assumes that the outcomes of the project are a good thing¹³. Such approaches have also heavily influenced the Tasmanian State Service.

On the other hand, there is an influential body of literature which accepts and actively promotes user involvement to the point that it is they who control the content of the change project. Mumford's ETHICS methodology is perhaps the most well-known example of such an approach, but while it has attracted interest, the approach not widely diffused normatively outside Northern Europe. Hoffer, George et al. (1996) suggest these approaches are not suitable in an American context which does not have

¹³ When asked about the drafters and the LSP Smyrk suggested that they were *utilisers* as well as users.

a tradition of organised labour as in the UK and Scandinavia. It is discussed in more detail in Chapter 4.4.

Cavaye (1995) terms these two distinct approaches to user involvement in systems development as:

- functionalism - where the involvement of users is sought to build correct and efficient systems; and
- neo-humanism - where participation is considered central.

The differences between these types of participation imply quite different approaches to the process of change. The former suggests the process is one of problem solving, where users can be viewed as part of the problem, while the latter implies extensive negotiation. These differences have substantial implications for the macro-processes of change, as pursued in Part 4. Most systems developers recognise the need for users to be involved, but this alone is not enough. Buchanan and Boddy (1992) cynically termed this neo-humanist approach the "truth, love and trust" approach to organisational change. Recognising the approach had many advantages, they suggest it should be used in conjunction with other major change agent activities of project management and the judicious use of authority and active negotiation. This is pursued in Chapter 3.5.

In general, although systems developers generally accept the need for user involvement, the management of this involvement has proved problematic. As Blackler (1992) comments,

Paradoxically, social scientists who have worked with design engineers... report that design engineers often *do* take observations about the inadequacy of their approach rather seriously. It is not, in other words, that engineers generally turn their backs on such questions altogether, dismissing the very idea of a user centred approach. In practice, however, even when they are sympathetic towards a user-centred approach, they usually do not display much ability to adopt one (p 286).

Beath and Orlikowski (1994) suggest that the general consensus in the IS literature is that user involvement is beneficial and likely to contribute to a system's success and the problem lies in the nature of the involvement.

3.2.3 Active and passive user involvement

People can be involved in a project in a number of ways and the nature of this involvement is worth examining. A distinction can be made between passive and active involvement. If someone is passively involved, they merely receive or provide information more active parties choose to give or request. The active parties actually decide the content of change while the passive ones only provide information or support. If someone is actively involved, they participate in the decision making process, actively contributing to the content of the project. Active user involvement leads to users' acceptance of the final system. This is not only because they gain

information on what the changes will involve and so do not fear them, but because they actively help shape the content of the project so that it will be acceptable to them. Mumford's ETHICS methodology strongly promotes active user involvement, so much so that it they who control the process and outcomes.

Passive user involvement pays lip service to the users and often only allows users to be informed and indoctrinated. Referring to Hafsi and Thomas's (1985) study of planning in Air France, Mintzberg (1994) makes the observation that communication can be very close to control. In Air France, discussion meetings seemed to have less to do with gaining input into the planning process and more to do with aiding the implementation of previously established plans. When users are passively involved, they are provided with information about the project or system and only provide input insofar as they correct "factual" mistakes. If users are passively involved, their opinions can be sidestepped and the macro processes of change can be quite different than if users are viewed as active participants. Their involvement is seen to lead to commitment as they are "educated" about the system.

Deconstructing a common and typical systems development methodology, James Martin's Information Engineering methodology, Beath and Orlikowski (1994) point out that the very term, *user* denotes passivity and can downplay their often professional status. The nature of the relationship is one between an expert and a client, where

...users are portrayed as naive, technically unsophisticated, and parochial, while the IS analysts are presented as more knowledgeable, more professional, and more corporate-minded. Despite the rhetoric of user involvement, the text has analysts exercising almost complete control over the development process and users playing a passive role...we realise this perceived dichotomy is not sustainable, in that the users, in the end, are expected to be responsible for the outcomes of the development process. This contradiction about authority over the development process and its outcomes reveals a deep confusion about the nature of the relationship between analysts and users, and particularly about the appropriate distribution of power, control, responsibility and accountability between analysts and users in the execution of a shared task (p 373).

As Beath and Orlikowski suggest, this contradiction can prove untenable as users are expected to be submissive throughout the process of development, while the systems developers, who have dominated the process, are expected to yield to the users when implementing the system. They suggest that some of the well-documented conflict between analysts and users can be traced to the nature of user involvement in systems development. The key to resolving this contradictory relationship is to examine and question this structural distribution of power, authority and control. Accepting users as active participants in the process of development would be a useful first step in this direction.

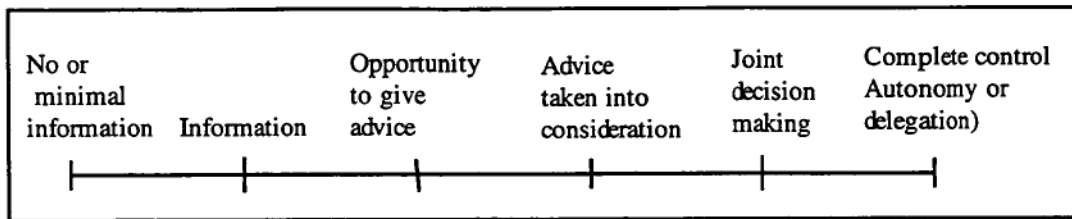


Figure 3.2.1: Scale of influence and Power in Participation (source: Heller 1992, in Krogstie 1995: p 165)

Heller (1991 in Krogstie 1995) observed that participation concerns the sharing of power and influence and devised a scale for considering the degree to which this happens through participation, as is illustrated in Figure 3.2.1. The more active the user, the more influence they have. Eason (1988) suggested there are three broad levels of user focus:

- technical centred - where users provide information required to produce technical specifications and are expected to accept the end products;
- joint user specification - where users are represented in all stages of the project; and
- user led, where the technical specialists merely provide a service to users.

Active user involvement in systems development projects with specialist systems developers involves joint decision making. Any more involvement would be classed as end user computing, while any less involvement implies the end users are passive players in a process largely governed by the specialist systems developers.

Friedman (1989) suggested that passive user involvement has become passé:

Participation has become less and less as a one-way communication forum from developers to users in order to overcome user resistance. Instead researchers have increasingly recommended user involvement as a two-way process, as a mutual learning process (Boland 1978) in order to improve the effectiveness of delivered systems (p 209).

However a review of recent literature, including the formal methodology employed by the LSP systems developers, suggests this trend is not as widespread as Friedman's comments suggest. Surveying a number of systems development experts in the UK, Clegg et al (1996) reported that while there was a general belief that user involvement was important, users were rarely influential in the actual design of new systems. Despite the general recognition of the importance of users' roles, the experts interviewed were critical of current practices. They attributed the continued lack of emphasis on users as attributable to a dominant technical orientation. As is pursued in Part 4, the commonly used macro processes of change do not adequately facilitate effective user involvement.

The overtly technocratic nature of systems development can influence how user involvement unfolds, suggesting why user involvement is often passive rather than

active. Technocratic rationality assumes there is, or can be, "one best way"; one which is derived from expertise. Burris (1993) notes that technocracy can interfere with democracy at a societal level; at an organisational level it can interfere with industrial democracy. For example, technical experts may conflict with group or user decisions because the "best" technical solution was not chosen (Ibid). She points out that these technocratic trends are being counteracted by models which, drawing their inspiration from recent participative social movements, redefine the role of an expert into one of a facilitator. In such a model, the nature of the professional/client relationship changes so that the dissemination of interests and meaning (or power and information, if you like) flows in two directions and mutual learning occurs. Experts become facilitators of this learning process and provide empirical or theoretical knowledge which is applicable to the particular situation, while it is the clients who decide on courses of action they are willing to follow. As Crowe et al (1996) stated,

The hours spent with the users to formulate definitions of articles and attributes... is not as it is sometimes arrogantly implied, about rationalising users' ambiguities and confusing terminologies but about arriving at a shared interpretation of the problem domain (p 106).

At a micro-level, active involvement implies extensive negotiation and compromise. At a macro level, it suggests the process is emergent, as discussed in Part 4. At an institutional level, it illustrates any changes are deeply embedded in their organisation context, being heavily influenced by people's perceptions which are largely shaped by that context. The difference between active and passive user involvement is thus crucial for later discussions.

3.2.4 User involvement and the LSP

The LSP team were generally aware that the OPC needed to be involved in the project from the beginning. While their formal methodology and initial expectations were that the users would be passively involved, they had to cope with sometimes very active involvement from the users who reacted negatively to their work. The LSP team realised they needed to cope with these reactions early in the project and put a considerable amount of effort into ensuring the project would meet the needs of the users within the constraints set by other requirements.

However, their formal systems development methodology did not provide effective support for them to do this effectively, suggesting there is a requirement for another approach which supports active involvement. The methodology recommended that all users should be involved in the early stages of the project but only passively:

...ensure that all management levels of the enterprise are aware of what is happening and why, either by memo or meeting. Information will be easier to obtain when people are aware of the reason for its collection. Always ask staff if they are aware of the reason for the study and explain very briefly. Memos and meetings do not always make things clear...

Having decided on a plan, *advise* the affected department so that their staff are forewarned and can prepare sample documents... [italics added]

The methodology explicitly warns the analyst and other systems developers to maintain objectivity. The negotiation of meaning is implied but the authors always assume there is one "right" answer.

Involve departmental representatives so that they are aware of your *findings*. Some of the information *supplied* may be wrong or you may have misinterpreted it. Find out early [*italics added*].

In practice, the CIPU's analysts and project manage had scope to use their own initiative and tended to only follow such methodologies when they saw the situation required it. However, there are indications there tended to be only passive user involvement promoted in the very early stages of the project, the prototyping exercise being indicative of this. This passive involvement may also have been attributable to some contextual issues, such as the lack of access to OPC staff members and the fact the CIPU were asked to develop the system by the executive manager rather than the OPC. There seemed to be some confusion as to whether they were enabling the users to develop a system or developing a system on behalf of them.

However, the LSP team often did promote active user involvement, and aimed to produce a system the OPC would be happy with. They were generally concerned that the OPC were not involved enough:

The OPC are not taking an active role (responsibility) for specifying their system requirements. The OPC will be required to "sign-off" the Functional Specifications as being an accurate representation of their system requirements. However, this stage is a little late for an effective contribution. The attitude seems to indicate a denial of change. The availability of the drafters' time is a constant issue. A phased implementation (of both technology and procedures) is emerging as the only plausible option (LSP Update 15/3/94).

If the LSP client groups do not have sufficient involvement in the project life cycle, then the resulting system may not meet their requirements (Business Case Document v 4 30/8/1996).

These concerns were also highlighted by consultants employed by the LSP team.

a) Consultants' reports

The systems developers employed two consultants to look at the OPC's needs in relation to computerised technology and general project management considerations around the time the functional requirements were being documented. They both recognised a need for members of the office to be more actively involved in the project.

In April 1994, a business consultant talked extensively with everyone in the office and produced a report describing the OPC's work and some recommendations for introducing computerised technology. Specifically, it highlighted that OPC staff did not feel involved in the LSP and lacked knowledge concerning computerised technology.

Most staff I interviewed are unaware of the abilities of the proposed system and do not feel involved in the decision. It is best to involve them now, before the computers are installed.

To involve the staff I recommend demonstrating the proposed hardware and software before it is installed...

It is best to deal with questions and fears earlier rather than later (Business Consultant's Report, April 1994: p 23).

By addressing their fears and concerns, the consultant was able to gain the trust of the OPC. The consultant was later employed to aid communication and take on some tasks which could not be completed by CIPU's very small staff.

Another consultant employed to aid the systems developers with project management issues also highlighted the ongoing need to involve the OPC staff:

Consultation for the development of the Functional Requirements has mainly involved the management of the OPC. It is essential that the participative process be expanded to include the drafters within the OPC who will be the major users of the proposed new system (Project Management Consultant's Report 11-12/4/1994).

The involvement of the OPC staff will be critical... Up to this stage in the project the OPC have put their drafting responsibilities ahead of the LSP. If this is not rectified, it will inhibit OPC ownership of the LSP and could create problems at the acceptance stage (Project Management Consultant's Report 2/6/1995).

I remain concerned at the inadequate level of support available for the LSP project from within the OPC (Project Management Consultant's Report 25/7/1995).

Work demands on the OPC during the current month have severely limited LSP Project Team access to the OPC... The ideal outcome is that the OPC is able to fully participate during the development phase and every option should be examined to allow this to happen (Project Management Consultant's Report 28/9/1995).

The systems developers were aware of the need to involve the users in the project, but had trouble gaining both access to OPC staff members and time with them. This was largely because the management of the office greatly felt ongoing pressure to produce legislation and did not believe the office had adequate resources to contribute to the project. The recommendations by the consultants not only highlighted the issue for the systems developers, but provided further justification for promoting greater involvement by all members of the OPC.

b) Information Session

On 20 June 1994, an information session was held for OPC staff by the CIPU. This meeting involved all the systems developers, the executive manager who had instigated the project, some of the contractors involved at the time and all members of the OPC. It was also my first contact with the OPC. It also signified more active involvement in the LSP process by members of the OPC.

The executive manager and chief drafter took an active up-front role in this meeting. The executive manager outlined the reasons and goals of the system, linking it with the government's social and economic policies. He repeatedly mentioned that cabinet "required" the OPC to develop the system. Presented in this way, the users had no

choice but to be committed to the project because, as public servants, they were required to follow the directions of the executive government.

The chief drafter outlined the government's aims in relation to the project, as per the Business Case document. He pointed out that, to meet those aims, the OPC would have to implement a drafting system for automatic consolidation and improve the form in which Bills are presented to Parliament, implement the new system and manage its impacts. The role of the CIPU was to "hold their hands". He and the executive manager both emphasised there would be no job loss due to the project.

The business analyst described how they thought the project would unfold and what tasks would be involved. While the broad goals of the system had been decided, the LSP team members were not sure of the details as they needed OPC input. She emphasised they needed OPC input and so encouraged active participation.

All the members of the technical design team were introduced as a group to the OPC and an informal lunch following the meeting was intended to facilitate communication between the different groups. The project management consultant commented that,

The information session conducted on 20 June 1994 appeared to be well received by all members of the OPC. It is most important that the existing positive attitude of OPC staff towards the project is maintained through ongoing briefing sessions and involvement of staff (20/6/1994).

During this meeting, several OPC staff members commented that they did not feel involved in the project. One staff member stated she was concerned that the resulting system would be as bad as the prototype. Following this meeting the manager of the OPC gave a copy of the functional requirements document to all OPC staff members to review in response to complaints that most people in the office knew very little about the systems development project. By facilitating active participation, the scene was set for the negotiation of meaning and interest which underlies the process of development at a micro level.

c) Discussions over the Functional Requirements Specifications

Although a considerable amount of time had been spent on it, the drafters' initial reaction to the draft requirements document were largely negative, reflecting not only differences between the systems developers and users, but also within the OPC. Three of the drafters wrote written responses to the document and their criticisms addressed several issues including:

- the language and concepts employed;
- the viability of the project generally;
- the drafters' lack of computing experience;

- the purpose of some or all of the functional requirements document;
- the impact of the resulting system on the process of drafting; and
- the systems developers' lack of understanding of the drafting process.

The drafters complained they did not understand large sections of the document and therefore felt they could not comment on it.

I had difficulty with a lot of the terminology. Whilst it is probably the kind of terminology that is in widespread use in computing circles, it seems somewhat contrived. For example - doesn't "degradation of system performance" mean "loss of efficiency"? I had trouble in "accepting" much of the jargon, particularly words like "functionality", "progression points" "functional decomposition hierarchy", "unarchival", performance issues", comprehensive incremental testing strategy", "relational database structure", "client server technology", "population of a production system" just to name a few (5/7/94).

Accustomed to communicating ideas in either a written or verbal form, they did not relate well to the data flow diagrams commonly used in systems analysis. There was some resentment that the document seemed to be written for computer experts while they had very little experience with computers. This breakdown in communication facilitated learning on the part of the users as they were introduced to computing terms and concepts. It also resulted in compromises, as the systems developers refined the document to include fewer technical phrases and concepts. In this way, the gulf between the users' and systems developers' frames of reference was reduced, at least partly.

The drafters could not initially understand the significance of some sections. One section outlined the operations involved in amending legislation and one of the drafters wrote in response:

I don't understand the significance of this at all. Do we have to do this at all. Do we have to do this before we start drafting/ as we draft? (5/7/95).

Another section described the basic structure and purpose of the tracking of documents in and outside of the OPC as drafts were developed into proposed bills and acts. A drafter wrote in response:

I have difficulty in understanding the practical significance and application of this Part. Prima facie it appears complicated and daunting but perhaps it is not of importance to the actual drafting process. I do not feel competent to evaluate it (5/7/94).

Such a statement is a clear indication of the different frames of reference the users had to the systems developers. As well as indicating the users' lack of understanding of the document, it signified the systems developers' lack of appreciation of the organisational context. The drafters were particularly concerned that important part of the drafting process had been overlooked by focusing primarily on automatic consolidation.

Unfortunately, the Functional Requirements Specification ... seems to ignore the drafting (or production) of legislation and to concentrate on what is needed to solve the second problem - access

to consolidated legislation. The overall feel of the specifications is that the [project] will set up what is needed for consolidation of legislation and that drafting must somehow fit into that model...

If this is the viewpoint of the [project team], the project is doomed to failure and you may as well settle for the prototype that has already been trialed because the drafters cannot draft for the purposes of consolidation. Drafting is a form of technical but also creative writing. To expect people to create in individual segments or to continually interrupt their creative process to file bits of the creation and enter other information in the file is to prevent the drafter from drafting. Our productivity will decline and we will become increasingly frustrated. There is not much point having an automatic consolidation process with no amendments being drafted to consolidate (5/7/94).

The systems developers had interpreted the structure of the drafters' work in a way the drafters did not relate to. The systems developers had seen it as three main entities which formed the core of their analysis:

- legislation- original acts, amendment acts, subordinate legislation and parliamentary amendments;
- segment - a generic name to refer to components within the body of legislation; any unit of information that may be uniquely referred to as a unit
- operation - an action that is performed on a segment for amending legislation eg. insert section; omit schedule (Functional Requirements Specification).

The drafters were particularly concerned about the analyst's interpretation of some of their work and members of the LSP team felt the users were blackmailing them with their threats to not use the system.

Drafters **do not and cannot** draft in segments...

If this is not taken into account in your system, the system will fail because drafters could not and would not use it.

Although I understand that legislation may need to be broken into segments for the purposes of consolidation, this Process called "Drafting of Segments" must be revised as it exhibits a complete failure by the authors to understand the nature of drafting (comments Functional Requirements Specifications 5/7/98).

Responding to the drafters' criticisms of the concept of segments, the business analyst maintained they actually were "coming from the same direction" but just used different terms for it or that she was labelling something they usually did not think about. The drafters disputed this and the business analyst amended the functional requirements to exclude all references to segments. This reflected a willingness to compromise, although it was included in related documents with more explanation.

The business analyst was very concerned about this negative reaction and felt in the "firing line". She felt they had not really understood the purpose and content of the document and held several lengthy meetings with the OPC staff, following them up with individual discussions designed to "increase the users understanding of the document to put them in a better position to review and comment on the document". That is, they still maintained problems lay primarily with the users' understanding of the situation, rather than their own. The resulting discussions prompted the systems developers to spend a considerable amount of time explaining the purpose and

meaning of some sections of the document and promoted the redrafting of some sections. The LSP team wrote:

The comments received from [the drafters] were appreciated and highlighted several important issues. However, given the nature of the document, it needs to be worked through in an interactive manner to gain the familiarity which is needed for review purposes (transcripts 12/7/94).

In order to explain some concepts, they produced another prototype at the request of the OPC.

The OPC requested the development of a second prototype (based on the Functional Requirements Specification) as a tool to help them to evaluate the document and endorse the requirements as an accurate representation of their needs (Review of Functional Requirements Specification 8/8/94).

The LSP team were quite concerned that this one could also be compared with the first prototype.

If the second prototype system is not perceived by the OPC to be an improvement on the first prototype then this could result in a considerable risk to the project (in particular the relationship between the CIPU and OPC)... The objectives of the exercise must be clearly identified and the scope of the systems development documented and understood (by both the CIPU and OPC prior to the development being undertaken (LSP status report 12/5/94).

This prototype (now called a "demonstration system" to differentiate it from the first prototype) was firstly shown to the chief drafter (15/7/94) before demonstrating it to other drafters. This demonstration system illustrated what was meant by computerised drafting facilities. At the same time, one of the drafters developed his own prototype, as discussed in Chapter 3.5. Prototyping proved a useful tool for facilitating active involvement, as the users were able to see what the system could look like and respond to it.

Following these discussions the business analyst produced what she termed a "plain English" version of the functional requirements which avoided technical issues and terms. The fact she used the term "plain English" illustrates a process of mutual learning. At the same time the LSP was being developed, there were discussions in the OPC, and legislative drafting bodies generally, about the benefits of using plain English terms rather than more legal terms or Latin in legislation. These discussions were not connected to the LSP project, but the business analyst's use of the term illustrates a growing awareness of the issues affecting the OPC staff.

Following these discussions, the functional requirements document was signed off and became an important document for guiding the tenderers as they presented their technical solutions to both the system development team and the users. The discussions over the functional requirements illustrate a mutual learning process whereby the technical systems developers gained a further insight into the requirements and the users gained an understanding of what the system could be like and what the process of computerisation involved.

3.2.5 Reflections: users as active participants

User involvement was an ongoing issue in the LSP, largely due to the fact the system was forced on the OPC and the heavy workload of the office. OPC involvement was cited as a concern throughout the three and a half years I tracked the project.

However, in the early stages, this involvement was to be primarily passive involvement. Although they expressed the desire for OPC staff members to be actively involved, when this involvement led to criticisms of their work, the LSP team were largely unprepared for it. They did compromise on some issues in response to the comments and criticisms of the OPC staff members. Nevertheless, even towards the end of the project, some of the drafters still felt that their requirements were being overlooked and no major changes were made to the functional requirements in response to the concerns they raised. Just before the system was to be implemented, one of the drafters commented to me by email:

It seems that we must adapt our thinking and procedures to accommodate the new program rather than to accept a program which can be utilised in pursuing our goals (transcripts 17/11/1997).

There were also concerns that opposition to, or criticism of, the system was viewed as ludditism or, as one drafter put it, “confirmation that legislative drafters... are adverse to change; that we are conservative, inflexible, frightened of (as opposed to unfamiliar with) technology etc” (email 16/3/1998).

The users in the LSP were not only passive providers of information but active participants in defining what the planned content of the project would be. This occurred even when the systems developers initially only promoted primarily passive user involvement. While the business analyst was obtaining information about the OPC and their requirements, OPC staff members were receiving information about what the proposed system could be like and were able to suggest what they would and would not like to see in the system. In other words, they were participating in an ongoing dialogue on what the content of the project would be. However, with the systems developers’ reliance on the common assumptions and conventions of the systems development literature and the fact the CIPU were acting as agents of change on behalf of the executive management rather than the OPC, this active involvement was not as great as it could have been.

Systems analysis and design can and should be considered a two-way, iterative and emerging process rather than a one-way, non-iterative process of extracting information. While systems developers are determining user requirements, users are discovering what the proposed changes may involve for them. Questions asked by the analyst in interviews do not just extract information, but provide information to the user and initiate negotiations concerning what is acceptable to the user when

changes are made. The problem with defining systems analysis and design as the extraction of information from users is that the political nature of systems development is ignored and users are relegated to a passive role of merely providing information. Active user involvement and the need to produce a technical solution which is acceptable to those who will use it strongly imply that the process of defining what the content of the project is greatly involves negotiation.

Recognising and promoting active user involvement has important implications for practice. An acceptance of users' active role in systems development projects means that differences in opinions between systems developers and end users cannot be simply relegated to resistance or ludditism. The assumption has been that involvement will almost automatically or directly lead to commitment. By involving users in systems development, they will come to "own" it. However, there is an important intermediary step. Active involvement implies extensive negotiation, as the proposed content, or outcomes, of information systems projects are discussed, compromises are made and authority and expertise are employed. This is pursued further in the following chapter.

3.3 Systems development as the negotiation of meaning and interest

Systems development can usefully be described as a process of negotiation of interest and meaning because this focuses on the dynamic and iterative nature of what is essentially a process of social interaction. While negotiation between systems developers and users is often suggested in the research literature and normative texts in the area, it is rarely explicitly examined. Here the process is explained using literature on negotiation, systems analysis and design and observations of the LSP. A distinction is made between the negotiation of interest and the negotiation of meaning and the intertwined nature of these two concepts are discussed.

Viewing information systems development as the negotiation of meaning and interests reflects the political nature of many information systems development projects and the importance of interactions between those involved. The need to meet users' requirements and embed the information system in its organisational context drives these negotiations, but the interactions surrounding such negotiations are not generally examined. Here the concept of the negotiation of interest and meaning is introduced and discussed.

3.3.1 Negotiated order

Strauss (1978) suggested that all social order was negotiated order as people interact. While much of the literature on negotiation focused on the resolution of conflicts, he pointed out that most social interactions involve negotiations as people discuss issues from different perspectives. For him, the process of negotiation is closely tied to perceptions of legitimation and related processes include persuasion, education, manipulation, appeals to rules of authority and coercion. The products of negotiation includes rules, agreements, contracts and so forth and the negotiated order at a single point of time could be viewed as the organisation's rules, policies and conventions.

Any changes to a social order, such as the arrival of a new member or changed environmental conditions, would promote changes in the negotiated order and this reconstitution of the social order could be seen as "...a complex relationship between the daily negotiation process and the periodic appraisal process" (p 6). Strauss identified the interconnection between negotiation, appraisal of existing social orders and the more "stable" rules and policies as a central issue, so illustrating the alignment of these concepts with Giddens' structuration theory. The promotion of a new information system is an example of a change in the negotiated order which constitutes an organisation. The process of introducing these changes can thus be usefully viewed as negotiation.

Strauss describes some of the sub-processes of negotiation as trade-offs, obtaining kickbacks, the achievement of compromise and the creation of negotiated agreements. These sub-processes only cover the negotiation of interests, but the negotiation of meaning is very closely related and the two are inter-connected. It is these processes of negotiation that underlie the systems development process at a micro level.

3.3.2 Organisational learning and the negotiation of meaning

Systems development has often been referred to as a process of mutual learning (Banbury 1987; Kwon and Zmud 1987; Land, Le Quesne et al. 1992), but this metaphor is rarely elaborated upon (Jones 1995). The problem with simply viewing IS development as learning is there is no recognition of the close relationship between frames of meaning and the interests of individuals and groups. The idea of mutual learning does not acknowledge or explain conflict. The negotiation of meaning does, and the similarity of the term to the "negotiation of interest" implies the close relationship between the two concepts.

The negotiation of meaning is the process by which people come to agree on common frames of reference. Different groups can have incongruent frames of reference while people who work together tend to share assumptions, knowledge and expectations. Social interaction and negotiation create opportunities for developing congruent frames of reference and exchanging different points of view. The negotiation of meaning refers to this process of the interaction and influence between people's frames of reference.

Orlikowski and Gash (1994) suggested that incongruent frames of reference surrounding the technology could create difficulties and conflicts during the process of developing and implementing systems, stating,

Early articulation, reflection, discussion, negotiation, and possibly change of inconsistencies and incongruencies may reduce the likelihood of unintended misunderstandings and delusions around the implementation and use of new information technology (p 202).

The systems development process involves the interaction between people with different experiences and background and part of the process of negotiation involves a mutual learning process. This process includes the agreement over what terms, phrases and concepts are appropriate and what they mean to all involved. Given the multi-disciplinary nature of many systems development projects, this negotiation of meaning is crucial for effective communication between all the parties involved in the process of development. Essentially, it is a fundamental element of the systems development process at a micro level.

3.3.3 Systems development and negotiation

Viewing systems development as negotiation contrasts strongly with the common structured approach to development. The structured approach constantly underestimates the difficulties associated with communicating with users, who are asked to "verify" logical designs (Bansler and Bødker 1993). Simply, structured approaches do not recognise or cater for the social processes surrounding systems development.

The negotiation of meaning is closely intertwined with the negotiation of interests as common terms of reference are established and the generally ambiguous areas of legitimacy are defined. Kling's explanations of negotiated order in web models suggests the intertwined nature of the negotiation of interests and meanings:

Negotiated order analyses focus on the ways that organisational practices, such as systems standards, are worked out between groups with differing interests and orientations (Kling 1992: p 393).

In other words, common meanings, such as systems standards, are created through negotiation. These common meanings could include more mundane issues, such as agreement over the definition of a word or phrase or the meaning of certain actions. In the LSP, for example, there was disagreement at one stage over what more technical terms such as "interface" meant and whether it was appropriate to use it in certain contexts. Definitions of words are often under debate in political discourse: as Truex (1993) points out, the debate over abortion is a debate over the meaning of the word "life" (p 30). Language discourse is not merely the implementation of existing structures of grammar and predefined vocabularies, but is "...negotiable in face to face interaction in ways that reflect the individual speaker's past experience of these forms, and their assessment of the present context, including especially their interlocutors..." (Hopper 1990 in Truex 1993: p 31). The negotiation of meaning is closely tied to an interactionist perspective on language.

Writers such as Kling (1987) and Walsham (1993) have long recognised that the organisational context of systems development is a negotiated order. Walsham's (1993) political analysis of organisations strongly suggests that negotiation is occurring, yet does not elaborate how political activity unfolds over time. The negotiation of meaning and interest is essentially politics in action.

There is a significant and growing area of literature which recognises the existence of differing perspectives and their interaction in the process of systems development.

Research projects have focused on, for example:

- mutual negotiation when both users and developers are aware that power is being exercised (Markus and Bjørn-Anderson 1987);
- descriptions of design as a process of learning, communication and negotiation (Curtis, Krasner et al. 1988);

- the "sedimentation of meaning" and the way that symbols gain a permanent presence in a health sector organisation (Prasad 1993);
- observations that finding the right conceptual model of a system is a process of "negotiation among various stakeholders", leading to "a basis for agreement with users regarding what will be built, created from a "combination of perspectives" (Moody, Simsion et al. 1995);
- observations of user-analyst interaction which suggest that both parties learn about the others' expectations and the gaps between peoples' expectations are modified through discussions (Hartzel and Flor 1995); and
- effective collaboration as an integral part of systems design (Gause and Weinberg (1989 in Crowe, Deeby et al. 1996);

Thus there is recognition that negotiation occurs. However, there are fewer studies focusing on how it occurs over the span of a project, as is achieved here.

The negotiation of meaning and interest is often implied but not explicitly examined. Many systems analysis and design and project management textbooks emphasise the need to, for example "establish a common language" between analysts and users (Martin 1995), the "management of stakeholders" and the importance of user review. Darke (1995) questions the general assumption that users do or should have a single objective viewpoint and suggests that an analysis of people's different viewpoints should be done, with one step of the process being the "management of conflicts and inconsistencies", after which viewpoints are "integrated". Most texts mention the need to identify users' differences in opinion and the need to "synthesise" their information needs (Jordan and Machesky 1990), but there is little recognition that analysts themselves play an active role in this social situation or how this synthesis is achieved. Words such as *integration* and *synthesis* are euphemisms for the political processes of negotiating an information systems project.

Most positively in this area, a number of recent textbooks describe joint application design (JAD) sessions in which all interested parties meet to define, analyse and produce prototypes of potential systems (eg Hoffer 1996, Dewitz 1996). This exercise forms part of the rapid application development (RAD) methodology which aims to speed up the process of systems development but significantly, it brings together people to discuss issues. The fact that negotiation would logically occur in these sessions is rarely admitted yet perhaps it is one of the reasons for JAD's growing popularity. JAD is discussed further in Part 4. Many more texts discuss the process in terms of gathering, analysis and problem solving, as discussed in the previous chapter.

A small but growing number of methodologies openly recognise that negotiation occurs. Perhaps the most well-known and established of these are Checkland's SSM and Mumford's ETHICS. SSM is described as "a participatory process because it can only proceed via debate" (Ledington 1989), though the process is likened to learning, rather than negotiation. Mumford's work expresses the concept of negotiation more openly:

All change involves some conflicts of interest. To be resolved, these conflicts need to be recognised, brought out in the open, negotiated and a solution arrived at which largely meets the interests of all parties in the situation... successful change strategies require institutional mechanisms which enable all these interests to be represented, and participation provides these" (1983 in Avison and Fitzgerald 1995: p 355)

However, these methodologies are not as well disseminated as the objectivist normative literature. This is possibly because they express a significant shift in thinking for many systems developers and are relatively complex to apply. Although the commercial methodology used by the developers of the CIPU mentioned SSM and they were introduced to some of the literature in the area, they did not employ it. Thus, there are approaches which actively support negotiation but, in general, texts only implicitly support it.

3.3.4 Negotiation in the LSP

Negotiation became an ongoing theme during observations of the LSP as people with differing frames of reference interacted to create a new system. The systems developers recognised they needed to address users' concerns if the system was to be a success, while the users were continually forced to be involved both by a concern that their work could be made untenable by external parties and by continual pressure from executive management for them to be involved. The process of developing, reviewing and signing off the functional requirements was an important time of negotiation. Here further incidents are described to gain an insight into the process of negotiation and meaning and interest during the project.

a) Change management workshops

With the aid of an external consultant, the CIPU organised a change management workshop to focus on the non-technical issues which seemed to be causing problems while the functional requirements were being finalised. The formal purpose of the workshops was:

...to identify individual concerns and needs with respect to the implementation of the Legislation Systems Project, and to discuss options to satisfactorily address these concerns and needs (3/8/94).

Thus, almost explicitly, this workshop was intended to be an open forum for negotiation, both between the CIPU and OPC and within the OPC itself. The first day-long workshop was held on a Sunday at a venue removed from the office to

promote a more relaxed atmosphere. The consultant acted as a facilitator and attempted to create a friendly, open and democratic atmosphere. All participants were invited to contribute to how the day should be run. As an introduction, the group, which included the OPC, members of the CIPU project team, the business consultant and myself, the OPC and CIPU were asked to identify all the players in the LSP. As well as reducing scepticism over the exercise, the identification of other stakeholders and later their concerns helped set the scene for the OPC and CIPU to discuss their concerns. The participants recognised that other parties had an interest in the success of the project and had to consider these concerns when articulating their own. For example, the needs of politicians and other government agencies were seen to include access to updated legislation. Their fears were perceived as including: disruptions to the legislative program during the development and implementation of the system, that the project might highlight problems with some statutes and that the new system may limit their ability to make amendments to legislation. In this way the concerns of the OPC and CIPU were set against a backdrop of a broader desire for the project to succeed.

The last external group to be discussed was the CIPU and the members of the LSP team were asked to outline their own fears and concerns about the project. With a flourish, the project manager pulled out a long list and read them out accompanied by groans and the rolling of eyes. With such humour, the project manager helped the members warm to the LSP team and relax in what until this stage had been a fairly awkward situation. Concerns of the project team included:

- that not ALL information needed had been gathered and that not all contingencies were planned for;
- that it would be impossible to please everyone and that there may be incompatibility between different groups; and
- that there was insufficient time commitment by the OPC and that the OPC would not change to meet the demands of the new system.

To overcome these concerns, they suggested they needed to develop a close working relationship with members of the OPC to obtain a "real feel" for the office's requirements. In this way, the meeting became a forum for negotiating issues such as access to the office.

The individual members of the OPC were then asked to consider their concerns. Some commonly cited concerns included:

- the worth of the project itself (that it was not addressing the underlying problem- the shoddy state of the statute book, for example);

- the ability of the new technology (with references to bad past experiences);
- that the new technology could have an undesirable effect on people's tasks and jobs (eg technology speed up, with one of the drafters commenting that the slower technology gave them a chance to think through drafting issues; job loss; a drop in job status or satisfaction);
- insufficient time or resources to plan and implement the system effectively;
- insufficient technical support once the system was implemented; and
- a lack of skills to help plan the system and use it, especially general computing and keyboard skills.

Commonly identified needs included:

- staff support / co-operation / a team approach / increased two-way communication / consultation / a positive environment in which to work;
- support and understanding from the systems developers and especially the executive government;
- adequate training in computing technology generally and the system itself; and
- clarification of future roles and career paths.

In a second change management workshop a month later, the group was asked to develop practical ideas for addressing these broad issues which the business consultant and analyst tracked. Specific training programs were established as well as other activities such as regular meetings to keep all OPC staff members up-to-date. These specific practical measures are discussed in other sections.

These change management workshops provided a forum for articulating concerns and negotiating aspects of change. While the broad goals of the LSP were presented as unnegotiable facts, decisions regarding how the OPC and CIPU should proceed and associated issues were raised and some of them were resolved.

b) Tender evaluation process

The systems developers decided to outsource much of the technical systems development in the project. Negotiations were apparent at this stage on many levels. A request for information was issued in September 1994, to which ten companies or consortiums responded. None of the short-listed companies or consortiums presented a pre-developed technical solution and this was considered a major risk. Due to the unique nature of the project the technical solutions presented either included large amounts of time for developing new software or adapting existing software or did not adequately address the system's requirements. This made it difficult for the LSP team

to determine which would be the best technical solution and to assess the ability of the tenderer to deliver what they promised. Hence, at the advice of a project management consultant, the LSP team included a further step in the tender evaluation process. They chose two companies which presented the most obviously feasible solution and paid them begin developing parts of the system. The aim of this "proof of concept" stage was to check they could deliver what they promised as well as gain further details on their proposal. For the purposes of confidentiality, these finalists will be referred to as X and Y.

X emphasised the need to spend time with the users to find out exactly what they required and stated that these findings could impact on the technical content of the system. They said their proposed solution was based on their interpretation of the functional requirements but emphasised that these interpretations could be adapted with further insights into the user's working requirements.

The environment we've chosen is an indicative one – it does not necessarily mean we will select it for the final project – we want to make that point – part of the project is devoted to getting the environment right (transcripts 21/2/1995).

The people associated with X seemed to have some empathy with the fears and requirements of the users. The team included a lawyer and the test documents used to show the users how the system would look were formatted to resemble Tasmanian legislation. Others in the team also emphasised the need to listen to the users and meet their requirements:

I understand there is a close relationship between you and the document [the draft] - it needs to be right - we can't let the technology get in the way of your requirements (transcripts 21/2/1995).

For some users this was the first time these fears seemed to be openly addressed by the technical systems developers.

Employee of X: The tools must not dictate your work. Your work must dictate the tools you use... and they should be easy to use.

Drafter (to me): That's good to hear.

Me: Had you heard anyone say it before?

Drafter: No, only that it would be easy to use (transcripts 21/2/1995).

Part of X's project plan included a phase where the drafters would choose an appropriate editor for the drafting environment. They seemed to listen to what the users had to say by, for example, stopping a demonstration of an editor which the drafters said they did not like.

The following week, Y were asked to present their proposal, using exactly the same format. They seemed to present their proposal as a solution for the technical systems developers and executive managers rather than the users.

Our proposal is a strategic resource that will satisfy tactical requirements... a conservative design based on proven concepts (transcripts 1/3/1995).

They made no effort to make the system look like the final system could look to the users, not even formatting test documents to look like legislation, unlike X. They made it clear they were only showing the technical viability of their solution and that issues surrounding the human-computer interface would be addressed later.

If there are any problems with implementation issues just remember its not really relevant – we're only showing the concepts behind it (transcripts 2/3/95).

The users were not greatly impressed with this attitude.

OPC administrative assistant: They didn't understand what we are doing.

Drafter: They just wanted to impress the project team.

OPC business consultant: Yes, they are the ones with the money, so they tackled them. They didn't see you guys as important (transcripts 7/3/1995).

Given the importance the senior managers and the systems developers had placed on meeting the requirements of the users and gaining their commitment, the focus of Y on the perceived requirements of the technical systems developers was misguided. Essentially, they were unprepared to negotiate.

Y's proposal emphasised the need to control and standardise the drafting process to meet the requirements of the technical system. They stated the drafters would have to structure their writing while they drafted and that they would only be allowed to use standard wordings.

Drafter: They have it the wrong way around - structure leading content - it's the wrong way around. They should think of content first then structure. Drafters work in different ways, but for me, structure comes later (transcripts 7/3/1995).

One of the drafters asked if she could type without having to structure the document but Y maintained this was not feasible because it could not be totally controlled. Another drafter stated that the structure of the document was not paramount while drafting. The first drafter afterwards commented Y over-emphasised the need for control and this concerned her. X had not done so, yet the LSP team's business analyst seemed to also think that such control was necessary and the drafter said she felt she did not know enough about computers to judge effectively. Y's emphasis on controlling the process of drafting to streamline it contrasted strongly with the drafters concern that the creativity required to draft could be stifled. Y described how the work processes within the office could be streamlined and would be governed by workflow technology but they failed to address these concerns.

Y spokesperson: If you come from a non-computing background, there will be changes, but it will streamline work practices (transcripts 1/3/1995).

It was clear to the users that Y expected to change their work practices to meet the technical requirements of the new system.

Administrative assistant: Many of the drafters have not used computers before.

Y spokesperson: There is no doubt it is going to be a change.. yes, there are mindsets... (transcripts 1/3/1995).

Again:

Y spokesperson: Education [of the users] is crucial for the success of this project.

In one of the meetings concentrating on technical issues, which the users did not attend, one member of Y literally stated that the users needed to be told what was good for them in relation to the most appropriate technology for their needs.

The users felt that Y also did not understand the drafting process and also that they were not willing to listen to the users to find out more.

Drafter 1: [X] were dedicated and interested in serving us, they were prepared to listen to us. [Y] said 'this is what we have, take it or leave it.' They said things couldn't be done, because we had seen them being done with [X].

Drafter 2: Whoever gets the contract needs to understand our world as much as we understand their world. I got the impression they just glossed over ours (transcripts 7/3/1995).

It was clear to the users that Y had made assumptions about the way the OPC worked and did not ask either the OPC or the systems developers to validate these assumptions. Y was not prepared to learn about the users and their work practices to any great degree. They also did not respond favourably to questions from the users.

Drafter [after the tender evaluation session]: One member of [Y] seemed a bit aggressive to my questions.

OPC Business Consultant: I thought they were a bit closed, a bit aggressive. They were close minded. That was out of hand.

Drafter 2: They invited us to ask questions and then closed up (transcripts 7/3/1995).

In other words, they were not prepared to negotiate. The users were concerned about Y's commitment not only to meeting their requirements, but the success of the system as a whole.

Administrative assistant: I thought they really had technical knowledge, but thought they didn't have interest or commitment and would not stick with us (transcripts 7/3/1995).

Y was a consortium of a number of different smaller companies and the systems developers and users were concerned they were not well coordinated as a group. Several members of the consortium had obviously not met before the presentations and different people sometimes gave conflicting responses to questions, reflecting a lack of co-ordination.

Drafter: The more they spoke, the more I felt they didn't understand how things fitted together.

Drafter 2: It was just a whole lot of packages fitted together.

Administrative Assistant: Yes. They said "Just trust us" (transcripts 7/3/95).

One of the administrative assistants said she felt X was more "user-friendly" than Y. The formal review of Y's presentations also reflected these concerns:

It is unclear whether all personnel who participated in the three days of tender evaluation meetings have a flexible approach to meeting end-user requirements (transcripts 22/3/1995).

While X had suggested a number of editors for the drafting environment and emphasised this would be an area where the drafters would have a large input, Y presented what they believed was the best solution. This concerned the users.

Administrative assistant: The whole thing hinged on the editor- they couldn't customise it.

Drafter: They kept saying this was the best editor, even when we didn't like it. They don't understand the way we think (transcripts 7/3/1995).

Several OPC staff members would have liked to have seen Y's proposal with a different editor, but had the impression they would not do so. These suspicions proved well-founded in later discussions.

After the initial presentations, the OPC were not satisfied with Y's proposal but did not feel confident enough to commit themselves to X. Y's technical solution was considered to be slightly better because it was based on more mature technologies. Hence, both companies were asked to return for further discussions about three weeks after the initial presentations.

The response of the two companies to the first presentations helped the OPC and CIPU choose between them. The decision was made by three groups including the OPC, as the major clients, a group of people with technical expertise in this area, and a group of people, including members of the CIPU and some external consultants, who focused on project management issues. While the OPC were the most favourably disposed towards consortium X, the other groups could relate to their concerns. X illustrated they were willing to listen to the users and systems developers by dropping the natural language parser included in the first presentation and had incorporated information gained from the three days of presentations into the detailed system specification. For example, they had changed the way amendments were carried out and revised the manner in which versions of documents were stored in the system. Y gave no indication they had considered the response of the users to their initial proposal by adapting it. They were not keen to consider another editor, stating that technically and ergonomically it was the best on the market. In other words, they were simply not prepared to negotiate.

Due to these factors, amongst others, X was declared the preferred supplier. There were also concerns that Y had not included key costings in their submission and would not break down the costings so the price of the system could be negotiated. However, given the importance attached to meeting the requirements of the users in this project, concerns about the relationship between the users and the tender company were a significant consideration. When the OPC were told that X had been chosen as the preferred supplier for the system, they expressed surprise that it could have been anything else. Their written assessments of consortium Y painted a picture

of technical systems developers who seemed unlikely to respond to the concerns of the OPC:

By the end of the sessions the staff concluded [Y] do not have an understanding of drafting and are reluctant to learn.

Some drafters felt, in the [Y] solution, the drafter is forced to work within a rigid structure, yet other drafters prefer to work this way. All agreed it is a personal preference....

The staff do not like the editor and felt [Y] showed reluctance to offer an alternative.

The OPC staff feel [Y] is oriented more towards the [systems developers] than to them as users... OPC are disappointed the tenderer did not bother to format the legislation to OPC's requirements...

Both companies indicated a willingness to *negotiate*. However, it follows that the Department would need to make significantly less sacrifices in order to reach *agreement* with [X] (transcripts 5/4/95; italics added).

The proof of concept tendering stage illustrates a number of levels of negotiation including:

- the formal negotiations between the tendering groups and the government bodies as they established what services were to be provided by each party at what cost;
- negotiations between the user, project management and technical groups, with each focusing on different issues; and
- the level of willingness each tendering group had in negotiating requirements with the users and their perception of the development process. The successful group recognised a continuing need to negotiate different issues with the users, while the unsuccessful group saw themselves as providing a technical solution to an already established and defined problem area.

c) Editor evaluation exercise

The LSP team decided to initially concentrate on meeting drafters' requirements in the drafting environment as they saw it as an important high risk area. In some respects the editor evaluation stage was like a prototype of the user interface for the drafters. It was during this stage that the drafters saw what the system could physically look like to them. It was also during this time that the systems developers began to understand the drafters need to think of the content of drafts before the structure.

The systems developers, who now included X's employees involved in the project, wished to ensure that the system's drafting environment was one the drafters would be happy to use. There were many discussions about what the drafters would like but for many of them it was difficult to make assertions as to what they would prefer when they had had very little exposure to computers. The systems developers also needed to choose an appropriate SGML¹⁴ editor to adapt to the technical needs of the rest of

¹⁴ Standard Generalized Markup Language (see ISO 1986)

the LSP, as well as the drafters requirements. Hence, time was set aside for evaluating available editors soon after X had been approved as the preferred supplier.

Learning from the experience of the first prototypes, the project manager emphasised to the users that the editors formed only one part of the system and that, if this part did not look suitable, it did not necessarily mean the rest of the system was inadequate. She emphasised that the SGML behind the scenes was the important element and that a number of different editors could be used with it. She reiterated it was the drafters who would choose the editor and that the editor they saw could be adapted to meet their requirements in certain ways. The project manager likened the editor evaluation stage like choosing a new car before knowing how to drive. The drafters started to laugh.

"Why don't you say, 'Just Trust us'," one stated. The project manager understood the joke.

"Yeah, like [Y]," she said.

X (hereafter called the technical systems contractors) evaluated over a dozen editors in terms of their ability to support the technical requirements of the system, the maturity of the product, the support for the product and its price. They narrowed down the choice to three due to these technical and price considerations. They had identified two different types of editor: those based on word processing packages and those which forced the user to be more focused on the structure of the documents by being conceptually closer to the SGML data structures. The first editor (editor A) was an example of the first class, while the second and third (editors B & C) were part of the second.

Editor A was based on the word processing environment already in use within the OPC and the systems developers thought the drafters might prefer a system which had a familiar interface style. The systems developers tried to demonstrate the editor by showing processes they thought the drafters might do with it.

We hope to show you a few things which we hope reflects the way you work (transcripts 17/7/1995).

They also tried to highlight any shortcomings they could see with the editor. For example, they had found that editor A was easier to use than the others, but that it was easier to make mistakes in the text which could not be effectively translated into the SGML data structure.

Editor B would not let the drafters make such mistakes but the users felt these restrictions would be highly constrictive when they were writing. This editor forced the users to employ SGML tags as they drafted and they could only draft using the structure of the predefined SGML data type definition (DTD). For example, they

would have to specify what part of the document they were typing before they actually wrote it. The drafters pointed out that at this stage they were more concerned about the content of what they were writing and that the structure enforced by the SGML tags and data types could be intrusive. This negative reaction to Editor B could be considered highly constructive as it promoted a greater degree of learning between the two groups:

Drafter 1: I think having to think about structure all the time, I can't think about content. If I had to do that, I might as well write it out by hand and then have [one of the administrative assistants] type it into the system. They can type faster than me. The way this [editor] is presented - all this extraneous information - it would break my thoughts too much. The one yesterday was not so bad this way, though it had some other problems

Technical contractor: But you use the structure like a skeleton.

Drafter 1: So I know how the document will look like when I'm finished, but this structure is often not put in until later.

Technical contractor: This editor makes you think of structure first, content second...

Drafter 2: I can't understand why a company doesn't come up with a package that lets you put in the structure later. There has been a fundamental schism between us and you, a difference. I just can't see the system working. Us the drafters and you the technocrats.

Drafter 3: Yes, there has been no meeting of minds... (transcripts 17/7/1995).

The technical contractor responded by saying it may be possible to have a template that can be filled in. The drafters liked this idea and further discussions revolved around what this template might look like and how the drafters could simply type the initial draft and then cut and paste it into the template. The drafters reacted favourably to this scenario. For example:

Drafter 3: When we draft we have to think about the content of the legislation and the format. If we had to think about the SGML as well, it would be too much.

Project Manager: The SGML is meant to replace the formatting.

Drafter 3: But it's too diverting for us. It would be better if we could just cut and paste.

Technical contractor: Perhaps you could just type a whole section and then cut and paste it into the format required.

Drafter 1: That would be okay... [Drafter 2 also agreed this would be much easier to work with].

Drafter 3: If you do the cut and paste method, you could avoid having to cope with SGML.

Technical contractor: We could do that with both the editors we've looked at.

Drafter 3 [to OPC manager]: If we had to think about computers and SGML and formatting and content we would end up with some pretty weird drafts [the chief drafter nodded] (transcripts 17/7/1995).

The technical systems developers still had to consider the technical issues which would be affected by the choices the users made, but they illustrated they were willing to listen to the users and take their viewpoints into consideration. One of the technical contractors stated soon after the above conversation:

The crunch will come when you want to save it as an [SGML] document. We'll have to think of that. It will come up with an error but it's not a problem (transcripts 17/7/1995).

This incident illustrated a difference in what the drafters thought they needed and what the systems developers thought the drafters needed. The technical systems developers were able to offset this difference and the potential conflict involved by presenting a slightly different scenario and by emphasising what had been shown was not the final solution. They made it clear to the users that they were only in the process of learning what the drafters required when the drafters reacted unfavourably to their interpretation of these requirements. They admitted that they had assumed that the drafters, like themselves, did not know what the structure of legislation was and so needed to be guided by rules, whereas the drafters were quite familiar with it. At the end of the meeting the technical contractor said he had learnt something about the process of drafting. He had assumed that drafters wrote a whole section before going on to the next one whereas now he realised that in practice they often jumped around between different sections, working on several or many sections at one time. Later in the project, he reiterated that he had learnt this directly from the reactions of the drafters. By doing so he illustrated to the users that he, a technical systems developer, was willing to listen to the ideas of the users.

The editor evaluation exercise is a vivid example of the value of perceiving significant parts of the systems development process as a process of the negotiation of meaning. The drafters learnt what the systems developers had interpreted their requirements to be and the systems developers had adapted these interpretations in response to the users' reactions. The OPC manager also recognised this at the time and stated so explicitly at the end of the meeting:

Project manager: We've been looking at each editor as it comes off the shelf... It's a very iterative process and we need your feedback. We've talked about templates today and so on. This is the first step, but it's not the final product. There'll still be a lot of input. We're only showing an early stage now.

Chief drafter: Yes, it seems you've gained information about what we want (transcripts 17/7/1995).

The editor evaluation exercise was a valuable way of negotiating what kind of interface was appropriate. However, several months later unforeseen technical problems led to the editor being replaced with a tool developed by the technical systems developers. The systems developers reported to the steering committee:

The editor that was chosen in July to form the basis of the drafting environment has been abandoned due to unacceptable performance when working with large sized legislative documents. The performance issue should have been identified by [the technical systems contractors]... (report to steering committee 13/12/1995).

As a result of formal tender negotiations, the technical systems contractors assumed responsibility for the additional work, which indicates negotiation occurred at a number of levels.

d) Detailed design

Most of the detailed design of the resulting system, EnAct, occurred in Melbourne, an hour away by aeroplane, and was consequently more difficult for me to track. User representatives were invited at times to visit the technical systems contractors' development site and detailed design sessions were scheduled so that the developers could obtain some early reactions to their work. However, geographic separation was an ongoing problem. A later review reported that a number of issues were not pursued with the OPC or CIPU and the technical systems contractors aimed for what was technically feasible rather than search for client approval, especially when they were under increasing pressure to complete the project (transcripts 24/2/97).

At times members of the CIPU and the OPC were asked to comment on work the technical systems contractors had completed. For example, all the drafters were asked to review early versions of the drafting interface soon after the editor evaluation exercise. This interface allowed drafters to enter information in two ways. They could either enter the structure of the document and then fill in the content in the relevant areas, or they could simply type in the content of the draft and structure it later. One of the drafters commented the second option was the one she would use the most as it reflected the way she worked, but she could see the point of the first way. She had just been to a drafting conference where this "top-down", or "forward" approach to drafting had been promoted. One of the Commonwealth drafters had created the sales tax legislation in this manner, including many key areas in large, complex tables. He had apparently spent ten months working on the legislation before writing a single word, but the local drafter commented,

That's okay for him to use this kind of approach to drafting, but the political reality for most of us most of the time is that we have to be able to show some progress (transcripts 25/8/95).

The other drafter agreed, stating,

It is very uncommon that the instructing officer has a clear idea of the line of policy and its structure. They only develop it at the same time as we are writing the draft and sometimes not even then- we have to write it for them (transcripts 25/8/95).

The two drafters commented that they finally felt the systems developers had finally understood the broad way the drafters worked:

It's taken a while, but I think they've finally understood that we don't draft in a "forward" way they thought - structure first, content second - but usually the completely opposite way (transcripts 25/8/95).

Parallels can be made between the drafting process and the systems development process and how the developers viewed the process of systems development seems to have coloured how they viewed the process of producing legislation. This is discussed further in the conclusions of this dissertation. What is important is that the

drafters continually negotiated on this issue and forced the systems developers to compromise.

Discussions and meetings at this stage seem to have focused on more detailed issues. The larger issues seem to have been resolved and now negotiations concerned more detailed issues, such as standard wording for amendments, the detailed design of the interface or the appropriate data type definitions. Much of the time, the OPC staff members would clarify what they meant by using examples. The technical systems contractors would then abstract from these. The detailed design sessions provided some good examples of negotiation over the more detailed system which needed to be resolved at this stage. The topics under discussion were almost too trite to provide concrete examples, but the following dialogue provides just one example of the ongoing negotiation which occurred throughout the project:

Drafter: We need to keep every version of a draft because sometimes the drafting agency may want to revert to how it was... Often we get questions of a philosophical or policy nature. For example, "why did we put that in".

Technical systems contractor: So you would need some kind of notation system.

Drafter: We don't now. It takes too much time. If we had the facility, though, we might use it sometimes.

Contractor: How do you differentiate each draft?

Drafter: By a number and date (gives example).

Contractor: You don't keep any information about the contents of the drafts - what changes have been made...

Drafter: No, not really.

Another drafter: There may be documents which are not part of a draft but are added in later.

First drafter: It would be good to see what differences there are between acts. That's why we discussed strikethrough/underline...

Contractor: Can you give some examples of when you would need to see the difference between drafts?

Drafter: Anything is possible. They keep changing their policies, they get advice and so forth... issues might become publicly unpopular so they change them several times, but they still go back to whatever version and often I don't remember what version it was.

Contractor: So you would like a list of when changes were made.

Drafter: Yes, that would be good.

Support staff: There have also been times when there have been different versions of the same draft bill at the same time.

Drafter: I call them "4" and "4a" or whatever. It also might not be the whole bill that is different. It might be just some parts of them and at the moment I put the alternative version in italics.

Another support staff: It has happened that two versions have gone to press and only at the last moment was the final version chosen.

First support staff: [One of the drafters not in this meeting] has two different versions of a Bill and keeps swapping things between them. Each of them follow completely different policies.

Drafter: We all have different ways of versioning them. We may have to develop some standard rules for this. Often I give the instructing agency part of a draft and keep working on other sections of it. We need to have some standardisation (transcripts 22/6/95)¹⁵.

Prototyping was frequently used during the design activities to give the users an idea of what they were aiming for and to check they had understood their requirements and were "coming from the same direction". Prototyping was especially used in developing the drafting interface, so that the drafters could effectively participate in discussions (or negotiations) concerning it. The CIPU project leader commented,

I believe that [the technical systems development] team has used prototyping very effectively to fill the design gaps resulting from the failure of 'off the shelf' software to provide an acceptable drafting environment (letter from CIPU project leader to technical systems contractors 6/5/96).

In other words, prototyping provided an effective forum for negotiating both interest and meaning.

Sometimes, though, the management of the OPC could make decisions and so bypass the process of negotiation through the application of their authority. For example, the technical systems contractors were spending a considerable amount of time producing SGML Document Type Definitions (DTDs) for the numerous different types of statutory rules. The many different types of statutory rules often differed in structure in only slight ways. The management of the OPC recognised that these differing structures added a lot of complexity and that it would be more efficient to standardise them, but initially believed it would be impossible. However, during discussions with the technical systems developers, they agreed it would be advantageous to reduce the number of different structures where possible (14&15/9/95). They reduced the number of data type definitions (DTDs) from over a dozen to two. This application of authority is discussed in more detail in Chapter 3.5.

At other times, the technical systems developers simply employed the drafting expertise of OPC members to gain information and detailed requirements. The new system would produce standard wording for amendments and the OPC had to decide what these standard wordings would be. One of the technical systems contractors commented,

We can have two ways of going about this: either we map things onto the way you do it now, or we change the way you do things. I have no problem either way (transcripts 22/11/95).

Negotiations then occurred within the OPC as the office decided what standard amendment wordings would be used.

Now and again discussions would return to broader issues such as the viability of the project or system or the utility of the system to drafters but generally these issues

¹⁵ This issue of standardisation is addressed in Chapter 3.6.

seem to have been resolved at this stage. At this stage at least one of the drafters recognised that their requirements would probably change once they became more familiar with the system. The technical systems contractor also realised this, but commented that they needed to work towards firm design proposals. Later on, when it sometimes seemed they had not been, one of the systems developers commented, "If they don't accept the system, they have some explaining to do. They signed the functional requirements document". This issue of set but inadequate requirements is important and indicates a need to recognise the emergent nature of the process. This is not achieved adequately in much of the normative literature on systems development nor formal practices. This important issue is pursued in Part 4.

3.3.5 Reflections: Negotiating meaning and interest

The LSP study provides some vivid examples of both the negotiation of interests and meaning and the closely connected nature between the two. During the project, the drafters were concerned that the quality of their working life would be constrained if the structure of legislation was strictly enforced at an early stage. By suggesting to the developers they would not use the system if it interfered in their writing process in this manner, the drafters were able to force the developers to compromise.

This is but one example of the negotiation of interests. There were several times when the users suggested they would not use the system if it were built the way the developers suggested. At the same time, the developers were aiming to meet the goals of the project as defined by executive management and the parliamentary government. They were able to force user involvement in the process of development so that issues such as those discussed above could be negotiated in the earlier stages of the project¹⁶. Despite their heavy workload and an lack of initial interest in the project by some people, the users had to become involved because of the strong importance executive management placed on the project and the systems developers used this as a bargaining tool. They had technical and operational requirements to meet and if they were not met, the system, and by implication their work, would be considered a failure.

The evaluation of the editor also illustrates the strong connection between the negotiation of interests and the negotiation of meaning. Part of the reason why the systems developers believed it was important for structure to be enforced early in the writing process was that they assumed the drafters had the same lack of familiarity

¹⁶ On reading an early draft of this dissertation, several drafters commented they had wished to be involved but were not given adequate opportunities. This may have been partly due to the management style of the office at that time, which did not encourage open communication and also simply a lack of understanding between the OPC and CIPU.

with the structure of legislation as they did. Negotiations over the terminology, diagrams and concepts employed in the functional requirements document centred on their meaning and appropriateness and facilitated a mutual learning process. While the users were learning about technical words and concepts, the systems developers were gaining a further insight into the worldviews of the users through the users' interpretation of their work.

These processes of negotiation were almost ubiquitous throughout the project. During the early stages, when the project was being defined, broad issues were resolved as the goals of the project were articulated and "signed off" as a guide for future actions. Later on negotiation focused on more specific issues, such as the wording for amendment legislation, the type of desk-top computers to be acquired and the layout of the screen.

Systems development can thus be described as a process of negotiation at a micro level. When proceeding successfully, it begins with the resolution of broad issues, such as the worth of the project and proceeds to more detailed ones, such as the functional requirements, the interface design and the detailed requirements. At times later in the project, the level of negotiation was occasionally widened again but, generally speaking, the topic of negotiation became narrower as the project progressed. When some of the users threatened to greatly "resist" the project later on, LSP team members delegitimised their actions by commenting that they had discussed the functional requirements document and been involved in its sign-off and that if they did not accept it, "they would have some explaining to do". Once issues were negotiated, they became the framework for guiding future actions. The resolution of these issues helped create coalitions of interest and meaning made up of people who broadly supported the project and is discussed further in Chapter 4.5.

Systems development is often described from a rationalistic stance, ignoring the political and social realities of the process as people interact to bring about change (Korac-Boisvert and Kouzmin 1995). Accepting that people have different perceptions of a situation and incongruent interests can easily lead to a focus on conflicts of meanings and interests. Yet it is the resolution of these conflicts which is important for information systems development success and the concept of negotiation focuses on this process of resolution.

Negotiations may have been more noticeable in the LSP than in other situations, but the brief survey of the literature earlier suggests its relevance to other cases. In this case study, many of the users had a strong legal background and were extremely articulate. Their work involved the critical analysis of ideas and this, plus the dialectical or confrontational nature of legal practices in Australia may have promoted

more open debate and discussion than is usually the case. However, the literature review suggests negotiation is also present in other similar situations and it is a useful concept to explain the social interactions occurring during the LSP. It explains how differing groups tried to promote their perspectives, how those promoting the system "sold" the concept to others, who in turn modified it by promoting their viewpoints.

Viewing systems development as the negotiation of meaning and interests helps to focus on the social processes surrounding systems development and the process of embedding the system into its organisational context. Web models and other types of context analyses effectively illustrate the relevance of considering people's perceptions, social interactions and the effects of existing organisational structures and roles on systems development but do not focus on how these influence the process of development. Basically, they provide a "snapshot" model of organisations. Building on such models, the concept of negotiation does not provide the same depth of contextual analysis as these models, but focuses on the process of change over time. The LSP provides many concrete examples of such negotiation while many previous studies have only addressed it conceptually.

So is it useful to think of systems analysis and design as the negotiation of meaning and interests? It is for a number of reasons. Firstly, it emphasises the dynamic nature of the social interactions surrounding systems development. Instead of relegating users to passive providers of information, it emphasises their active involvement in the process. Involvement can often breed commitment (Amoako-Gyampah and White 1993), and users' commitment to a system is important for success.

Secondly, recognising systems development as a process of negotiation should result in more achievable (implementable) systems and may improve the rate of information systems project success. Negotiations surrounding the systems development process helps embed the system in its organisational context and so links the planning and development of information systems to their implementation in the organisation.

Thirdly, in redefining the process of systems analysis and early design as negotiation, novel and possible fruitful research paths are suggested. Conceiving the process as the extraction of information and the formulation of a technical solution suggests that, if there are problems, they are likely to be due to inadequacies in "mining" the information from users and converting it to a technical solution. Focusing on the process as negotiation suggests research should not just concentrate on the effectiveness of the one-way flow of information from users to systems developers, the developers' ability to transpose that information to a technically implementable solution and the effectiveness of training programs for users. It should also focus on the ongoing interactions between the parties involved in the process of development.

This has implications for practice too. If the early stages of IS development is viewed as negotiation, problems can be redefined as the issues of conflict resolution, a lack of bargaining ability and the inability to learn about other parties' perspectives. Solutions to these problems would focus perhaps on the interpersonal skills of all parties involved in the process of development and the facilitation of adequate and appropriate interaction. Many management tools used by systems developers, including those used in the LSP, do not explicitly recognise the highly iterative and negotiated nature of systems analysis and design process, suggesting further research questions: do they help or hinder the process of negotiation? What is their use in the negotiation process? Many tools and texts do not adequately discuss issues of conflicts of interests and meaning, relegating such conflicts to "a lack of training", neo-Ludditism or "resistance". This may be useful for justifying or rationalising systems developers' attitudes and actions, but will it help the process of creating new successful information systems if a key success factor is user acceptability? This indicates a need to examine perceptions of the macro processes of systems development, and as is done in Part 4.

Redefining the process of information systems development as negotiation is useful for both researchers and practitioners. By giving us a slightly different perspective on the situation, it suggests new and possibly fruitful avenues for research. Viewing information systems development as negotiation of interest and meaning provides an insight into the iterative and highly political nature of much system development activity by focusing on the social interactions of those involved. The following chapters focus on particular aspects of this process.

3.4 Creating and sustaining coalitions of commitment of support

Systems are created by people operating in coalitions of meaning and interest and these coalitions need to be created and sustained. In other words, those responsible for a systems development initiative must create and sustain a coalition of commitment to the initiative and its proposed outcomes. Here, the creation and sustainment of such a coalition is examined, with illustrative examples from, and analysis of, the LSP.

A systems development initiative must have some level of support if it is to be adequately achieved. Adequately defining a project is essentially the creation of a coalition as those involved come to an agreement on what is appropriate and achievable. If a project such as the LSP does not have the necessary support from those providing resources and those who will be utilising the system, it is unlikely to be successful. The creation of a coalition of interest and meaning for the project as a whole is one key area of negotiation and is examined here with illustrative example incidents from the LSP. In essence, the creation and sustainment of a “coalition of interest and meaning” are the end result of effective negotiation and support successful systems development.

3.4.1 Negotiating technical change

Creating and sustaining commitment to proposed changes is crucial. As Mintzberg stated,

Intended strategies have no value in and of themselves; they take on value only as committed people infuse them with energy... (Mintzberg 1994: p 172).

Coalitions of commitment need to be established and to facilitate them, policies are often defined in vague terms to broaden their support base (Majone and Wildavsky 1978).

This need to create coalitions of interest and meaning has become more important as information systems have come to be seen as strategic directions which can greatly change the way an organisation operates. Concepts such as business process reengineering (BPR) have highlighted the need for organisational changes to be associated with technological changes (Hammer 1990). Yet Brinckman (1991) argues that technological change projects as they are generally conceived, are incompatible with organisational changes. This is because there is a fundamental contradiction in the logic of successful organisational innovation and the requirements technological change projects. Organisational change is not a result of achieving

predetermined goals but of compromise, while technological change programs assume that such coalitions have already been achieved. This is a great oversight.

As discussed above, the systems development literature generally does not describe the process of defining a project as the creation of a coalition of meaning and interest, but as the analysis of a problem situation. The often mentioned importance of user review and top management commitment insinuates that negotiation is occurring, yet the process by which this occurs is rarely examined.

3.4.2 Marketing and obtaining commitment

"Marketing" a project is important. Observing the implementation of a large information system in the health sector, Brown (1995) observed that the project team marketed their actions in order to legitimise the system. He found they tended to give different information to different groups, emphasising issues they believed their listeners would find more persuasive. In this way, they attempted to create coalitions of commitment. Brown termed this "establishing the legitimacy of the IS" in order to create commitment.

However, there is a distinction between marketing and creating a coalition of interests. If we simply market a system, we may not recognise that others have alternative, but equally valid meanings and interests – they only need to be convinced that ours are valid. Viewing the process as the creation of a coalition of interest and meaning recognises that multiple and possibly conflicting but equally valid frames of meaning and interest are involved. Marketing then becomes a tool that is used to educate, convince, or indoctrinate others into joining a coalition.

3.4.3 Tactics for sustaining commitment

Creating such coalitions of meaning are not just crucial while planning and defining a project. Sabherwal and Elam (1996) observed that the successful development and implementation of an information system requires that the many problems encountered on the way are resolved. They suggest the resolution of many of these issues requires the "building and sustaining of commitment" by those involved, including all stakeholders. The fact that commitment is required for systems development to be successful, and that not having it has been a major cause of problems in developing information systems, though, is a truism. While most prior research had simply tried identify what factors were involved (such as the behaviour of people involved and resources), Sabherwal and Elam focus on the tactics used to build and sustain commitment.

An example of such a tactic for creating and sustaining commitment identified by Sabherwal and Elam included the involvement of key stakeholders " to instil a sense

of ownership". Focusing on a single retrospective case study, they identified several others, including: seeking out champions, periodically demonstrating the system's value to senior executives, publicisation and demonstrations. They effectively illustrated how these tactics were associated with the resolution of specific problems during the project. For example, they linked "seeking out champions" and "periodically demonstrating the system's value to senior executives" to the perennial problem of resources for what was an expensive project. Thus, a number of tools for negotiating commitment to a project can be identified, including marketing, the involvement of senior executives as project champions and participation by all stakeholders.

Generally, though, these tactics for creating and sustaining commitment are not explicitly examined or are expressed in euphemistic terms, such as the need to "sell" the project to the users, "create user awareness" of the issues or the need to "promote top management support". Observations of the LSP echo those of Sabherwal and Elam and give another example of how these tactics of creating and sustaining commitment can unfold in action.

3.4.4 Obtaining commitment to the LSP

The systems developers and executive manager responsible for the LSP recognised the need to "sell" the system to those who could influence it. The potential users and their clients had to be aware of the planned benefits of the project so they would not begrudge the problems and hard work associated with implementing it. Cabinet and the government bodies responsible for approving finances and authority had to be convinced if they were to provide the resources for the project. They would be more likely to do so if there was broad support for the project, so it had to be "sold" widely. This marketing was an integral part of the project. Through it, and the involvement of key people in planning sessions, coalitions of interest and meaning were created.

a) Obtaining the commitment of Cabinet

Funding had to be approved by Cabinet and, in order to obtain this funding, they had to be convinced of the project's viability and value. In March 1994, the LSP team worked on a submission to Cabinet and budget committee. Most of the submission gave broad justifications for the project, basically as described in Chapter 2.1. Cabinet's support was essential for the project and those promoting the project had to effectively market it to them.

The project's costs were also mentioned but not overemphasised in the attempt to sell the project to those controlling the resources. The executive manager advised the project manager to,

Try not to overstate the costs but at the same time, if the asset is to be protected, we need to identify enough to do a decent job (email 30/3/94).

Cabinet and the Budget Committee were convinced of the project's value, for the reasons described in the business case document, and provided funds for the project to proceed. In fact, the then Premier was so convinced by the planned and potential social and economic benefits he wished to publicly launch the project himself. In doing so he illustrated that the actions designed to "sell" the system had worked.

In turn, the Premier held a press launch and reported that the total cost of the system would be \$1.84 million, but justified it by referring to the estimated quantifiable and qualitative benefits:

It has been estimated that the consolidated legislation database will lead to a total reduction in input costs of at least \$1.3 million per annum.

These savings will be achieved by the major users of legislation, including the legal profession, business, municipal councils and Government Departments.

At the end of the day this means savings for the people of Tasmania, for several reasons...

- It will encourage greater competition in fees charged for legal services.
- It will reduce indirect business and government costs that would otherwise be passed on to consumers.
- Finally, and this is of relevance to the legal profession and their clients, it will reduce the cost of direct time-based charges.

I believe these reforms are another important step in improving Tasmania's overall economic competitiveness and efficiency.

But the Legislation System Project will also mean that the community will now be able to access the law in a more affordable and understandable form, either by purchasing a consolidated statute or by using our sophisticated library system.

Our Project also delivers other major social benefits to the Tasmanian Community.

Obviously properly consolidated legislation will improve the effectiveness of Parliament and the legal system (15/9/94).

Using the above justifications, the Premier aimed to create a public coalition of commitment to the concept of the LSP generally. Again, the Leader for the Government in the Legislative Council had been particularly involved in the project, and reported to the press at the same time that:

This is one of the most important law reform initiatives Tasmania has ever seen. Not only will it improve access to law by all Tasmanians it will also improve the operation and effectiveness of our Parliament....

The Legislative Council is the House of Review. Its primary role is to evaluate legislation forwarded to it from the House of Assembly.

And yet in the past, we as members have found it extremely difficult and time consuming to determine precisely what the present law actually is. Obviously this is something we have to do before we can even consider examining suggested amendments to that law.

If Parliamentarians who are responsible for making the law are having difficulty in accessing the law then imagine how hard it for everyone else (15/9/1994).

By marketing the LSP in this way, the politicians were able to justify funding the project. The systems developers felt under pressure to perform, given the greater

level of public scrutiny such publicity yet, at the same time, such publicity helped obtain the commitment of the OPC and others by illustrating that the parliamentary government (whose directives they were to follow) were very committed to the project. Thus, this public launch helped creation a coalition of commitment to the project on a number of levels.

b) Promoting users' commitment to the project

However, it was not enough that there was senior management and government support for the project. The resulting system had to be acceptable to the users, particularly the drafters. The drafters, and to a lesser degree, their support staff, could not be sidestepped due to their knowledge of the area and a general lack of available external expertise. Their commitment to the project was essential.

The LSP team were in a difficult position. They were obliged to produce a system that was acceptable to the users when the users were not particularly predisposed to the specifically proposed content of the project. While the staff of the OPC broadly accepted the aims and objectives of the project and the need to consider the benefits computerised tools might provide, they were not convinced that some aspects of the project, such as a requirement for automatic consolidation, were in their interests or the interests of effectively producing legislation. Yet the CIPU were obliged to produce a system with these attributes and were in a bind. They were required to produce a system with automatic consolidation and a computerised drafting environment which was acceptable to the drafters, when the drafters believed these two requirements were, to a large degree, incompatible.

Initially they asked the executive manager who had promoted the project to force the OPC to comply. This compliance was obtained to a degree, but LSP team became aware that it was important for the drafters particularly to be committed to the system and the development process if the project was to be a success. They and the executive manager realised they needed to market the system to the users and held several meetings with the management of the user organisation and the whole office. Even after the functional requirements had been developed and the change management workshops, the project director believed the OPC management and staff sometimes simply could not see the point of all or some of the project.

Suspecting that the members of the OPC had trouble conceptually linking the LSP with the goals and objectives of their office, the project director initiated a review of the OPC's goals and objectives and how they linked with the LSP in September 1994. This review exercise was based on the recommendations of a well-accepted paper in this area (Wetherbe 1991) and the project director asked the business analyst and a

business consultant to take the drafting office through the steps and concepts outlined in the article.

The analyst and the business consultant did not see the point of doing this so late in the project, but felt compelled to follow the directives of the project's director¹⁷. To the OPC, they justified the meetings by referring to a key concept from the Wetherbe article:

Information technology projects must be aligned with the business objectives of key client areas (transcripts 20/9/96).

"So", said the analyst, "we need to go back to the beginning". They also suggested the purpose of the meeting was to check the analyst had all the "required information".

The project manager and business analyst led the members of the OPC through a discussion about what the objectives of their office were, problems in achieving these goals and how information (and the LSP and computers generally) could help overcome their problems. The discussions linked the LSP to the objectives of the office and problems people had in completing their roles, but it is difficult to assess whether it helped make those connections in the eyes of the users. Certainly, the aim at this stage was to obtain the commitment of the users to the project. The participating systems developers commented they received very little information out of it. Success in obtaining user commitment was possibly undermined by the fact this was held relatively late in the process of development and users could not see the point of the exercise itself. It does illustrate, though, an ongoing perceived need to obtain the commitment of the key users to the project.

Believing that it was important for the members of the OPC to remain informed about the progress of the project and have some forum for contributing to it, regular meetings were held for the office during a large part of the project. Held usually on a fortnightly basis, the meetings began soon after the drafters responded to the functional requirements document and around the time of the change management workshop. Initially they were run by the business consultant but were later taken on by the project manager. The meeting provided ongoing information about tasks and covered quite minor issues, such as ongoing decisions about the choice of font the office would use with the new system and problems the office were having with the new in-house printer.

The LSP team used these meetings as a forum to gauge individuals' perceptions of the project and system and to respond to them. For example, a staff meeting in early March 1995 was planned to:

¹⁷ The application of authority is addressed in Chapter 3.5.

- Emphasise that drafters will not be forced to use the drafting system and will have the opportunity to migrate (if at all) from the manual system at their own pace.
- Emphasise that the new system will not be introduced this [financial] year (and that it is recognised that the OPC is under a great deal of pressure with the legislation program for the current year).
- Address concerns that the project is unfeasible (ie Tasmania can't lead the way etc).
- Emphasise that the LSP is a Government initiative, and that the OPC and the Parliamentary Clerks will be required to adopt the new system and new processes (ie, no choice) (Project management meeting 8/3/95).

Towards the latter part of the project, the CIPU believed that these meetings had become ineffectual and that some members of the OPC resented this ongoing and perceived useless intrusion in their time. As the project became delayed and stuck in development and testing activities, there was also less to discuss. In essence, the meetings were no longer an effective mechanism for creating and sustaining commitment and in fact seemed to interfere with it. At this stage, responsibility for these meetings was given to the management of the OPC.

This issue was discussed in some detail. It was agreed that although successful measures had been put in place in the past, it is now time for OPC management to take responsibility for ensuring the information relating to the LSP is made available to the rest of the Office, rather than the LSP team. It was suggested that the Acceptance Testing team could have 'round table' meetings with the rest of the Office to keep them up to date (OI FM 320, 1/10/96).

The CIPU were in a difficult position. When they conducted meetings, the users resented the intrusion into their time and did not appreciate being informed of minor decisions. However, when they did not hold meetings early and late in the project, they were accused of not keeping the users informed. This illustrates the difficulty of creating and sustaining commitment.

Although not really part of the LSP project, the CIPU also provided some technical support for the OPC. As they worked out what the requirements were for the new system, it became apparent that the OPC had suffered from a lack of technical support for quite some time. Technical problems at the start of the LSP ranged from no access to service-wide communications networks, hence no electronic mail (potentially very useful for facilitating contact between the OPC and their clients), a lack of training in wordprocessing skills, problems with the in-house printer, no systematic upgrade of existing software to inadequate backup facilities. Even though such activities were outside the scope of the project, they provided a number of benefits. It improved communication between the OPC and CIPU. They could give the OPC better access to up-to-date technology and hopefully make them more aware of its potential. Also they could provide some concrete services for the OPC when the visible outputs and outcomes of the LSP were not to be achieved for some time.

The LSP team were aware they needed to gain the trust of the users in both themselves and the technology and saw the provision of such services as one way of

facilitating it. First, the CIPU's technical officer provided such services, and the business consultant organised for the administrative assistants to receive specialised wordprocessor training (see Chapter 4.3). The technical officer was aware of her role in aiming to facilitate greater user acceptance of the technology and consciously tried to promote interaction and questions as she installed new software and so forth. Later the CIPU facilitated more permanent technical support for the office by promoting the appointment of an officer who would be responsible for the OPC in the longer term. Before the implementation and "handover" of EnAct, he aided the CIPU in developing the system. This technical support officer left as the system was being developed and ongoing technical support was handed onto the department's IT support section.

Thus the LSP team employed a number of tactics to induce user commitment to the project. These tactics included enrolling the OPC in discussions which linked the LSP to the purpose and work processes of the OPC, regular meetings and the provision of technical support.

c) Presentations for stakeholders

In addition to obtaining the commitment of the key users, Cabinet and the Budget Committee, those involved in the project recognised they needed to obtain widespread support for the project. This was at least partly because many members of the OPC were concerned that their clients would not appreciate delays in the production of legislation while they focused on the project.

There were stakeholder briefings for all parliamentary parties, the Legislative Council, senior private secretaries, magistrates, parliamentary clerks, Hansard representatives and parliamentary librarians, judges and agency representatives by the executive manager responsible for the project. The minutes of the steering committee reported that,

To date the briefings have been well received and have generated considerable interest in the project (Steering Committee meeting #4- 22 Nov 1994).

Significantly, these talks were very much marketing exercises. All possible benefits from the proposed system were highlighted and benefits which could flow from the system, but which were unlikely to be a direct part of it, were emphasised (eg reduced time before the courts due to better legal researching and access to correctly "pasted up" legislation, reduced government costs due to efficiency gains in the legislation production process and return from value added products, social policy reforms, such as improving the effectiveness of parliament). Milestones achieved in the project were stated and plans for the rest of the project were presented. The presentation concluded with the short demonstration of what some of the value-added products

from the project could be. The scope was later retightened so as not to include some of these benefits upon which the system had been sold as the project was at risk of becoming unwieldy due to a “creeping scope”, but it did help obtain commitment to the project at this stage. The project was also effectively linked with the broader socio-economic reforms being promoted by the government, such as improved access to the law.

The OPC members attended the presentation given for some of their clients, the agency representatives. At the beginning of the presentation, the chief drafter was introduced as “the guiding light” of the project before the executive manager went on to outline the impetus and reasons for the project, as outlined in the Business Case and chapter 2.1 here. During this meeting, as in the others, the LSP was effectively linked with some of the broad social and economic policies of the government.

Given the project was described in this way, the OPC really had no choice but to be involved and committed. This was because:

- The commitment of the government (Cabinet) for the project added a further level of authority for requests for the commitment and time of the users. The purpose of the public service is, of course, to operationalise the policies of the executive government (via Cabinet), so forces promoting the project gained a measure of authority;
- The publicisation of the perceived benefits of the system made the users aware of the possible longer-term impacts of the system and increased their perception of the importance of the system in the user organisation;
- If they opposed the project, they would be implicitly opposing these benefits to the community and other government bodies;
- In presentations to other stakeholder groups to publicise the system, the user organisation and specifically its managers were described as the custodians of the new system, making the users aware of their responsibility to the new system.

These factors helped gain the commitment of the users to the new system and the development process and, while they continued to have a heavy workload, they became more involved in the project.

3.4.5 Reflections: Commitment and a systems development project

Real-time observations of the LSP suggest that the definition of the project was just as much the result of negotiations leading to a coalition of interest and meaning than the gathering and analysis of facts. This suggests that, while Brinckman's (1991) observations are pertinent for much of the literature in the area and the formal

descriptions aligned with them, they may not reflect the actual day-to-day actions of systems development projects. Observations from the LSP suggest that many of the actions designed to "sell" the project helped build a coalition of interests and, through these marketing exercises, the scope of the project was slightly redefined. This stands in contrast to much of the literature on systems development.

A number of tactics were used to build and sustain coalitions of commitment as part of the planning, development and implementation process during the LSP. These observations largely reflect those of Sabherwal and Elam (1996). Using their analytical framework, tactics employed during the LSP are outlined in Table 3.4.1. Such tactics were particularly important for a project strongly promoted by executive management and almost enforced on the people who would be the prime users. No matter how the project was marketed as beneficial to the OPC, the CIPU became aware that, apart from the provision of word processing technology and a database of consolidated legislation, significant components of the system were seen by members of the OPC to primarily provide advantages for the Service as a whole and not the OPC.

Only some of these tactics have been addressed in this chapter. Given the strategic potential of the systems, its interorganisational nature and the broad social and political implications, the system was not only a high profile one, but potentially quite complex, and the systems developers were aware that they had to conduct the project to high quality, recognised standards, and be seen to be doing so. The social and political implications of the system were also used to "sell" the system to the prime users of the system. Continued user involvement in reviews and testing and decision making meetings helped, not only to determine the sometimes complex requirements of the system, but also to facilitate user acceptance.

User acceptance of the system and computerised technology itself was an ongoing issue, and enrolling the active provision of the chief drafter and executive management, training in and exposure to computerised technologies plus ongoing technical assistance were used to offset this potential problem. The users were unable and sometimes unwilling to spend time on the project, due to their heavy workload, and the CIPU actively tried to schedule activities which required their active involvement around these commitments as much as possible.

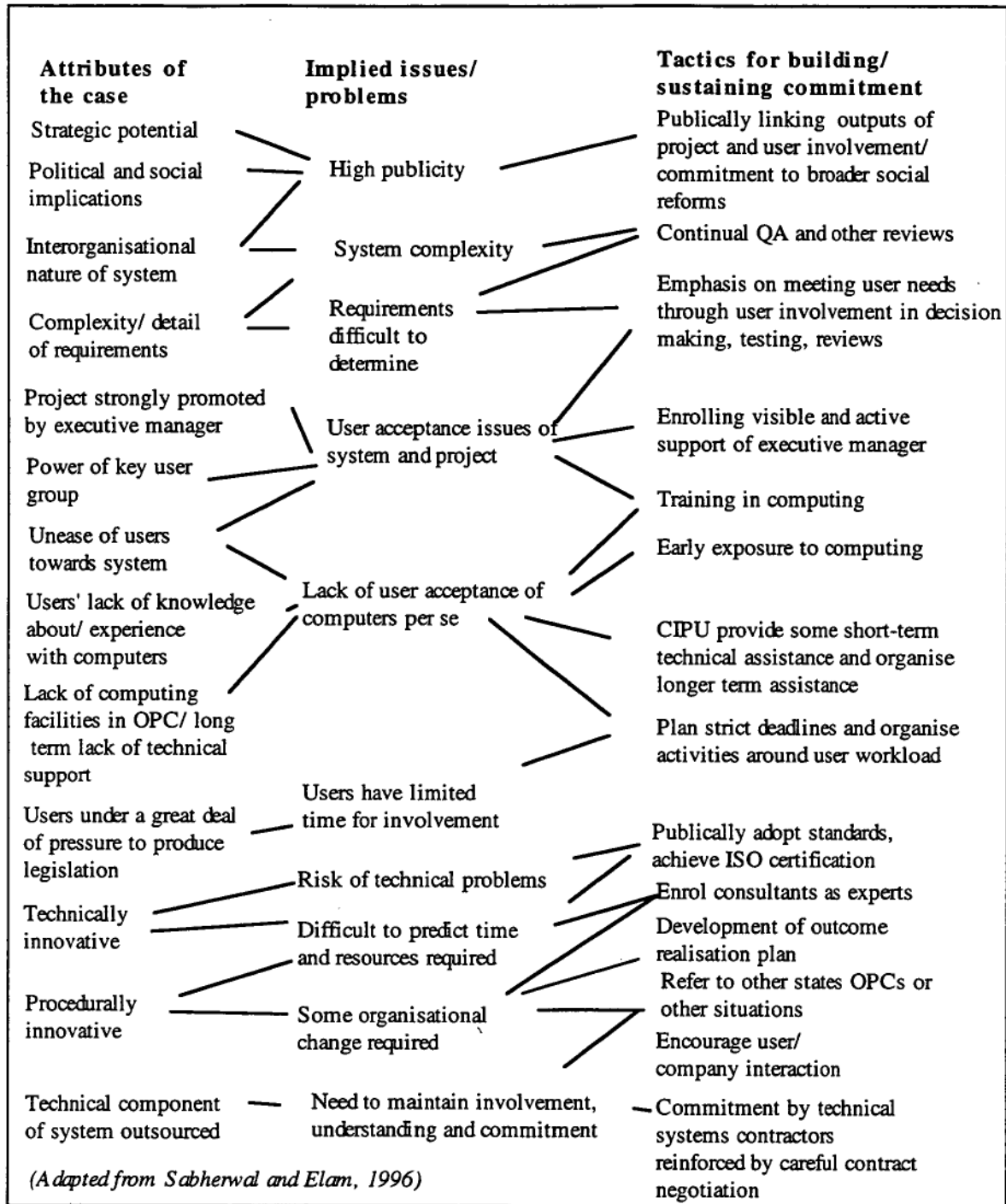


Figure 3.4.1: Tactics for building and sustaining commitment to the LSP

The project was technically innovative and so involved potential risks. These risks were realised, as the system proved unexpectedly technically complex (see Chapter 4.3). One tactic that was used to offset this problem was to publically adopt standards. They achieved and sustained consistency with ISO standards during the LSP and adopted technical standards, such as SGML.

The system was also procedurally innovative and so some organisational changes were required. The LSP team referred to other states' experiences, documented

existing procedures and actively tried to promote OPC consideration of this issue throughout the project. Responses to these issues were eventually to be outlined in an *Outcome Realisation Plan*, as discussed in Chapter 4.3.

These tactics are often referred to as change management, an issue pursued further in Chapter 4.1. As discussed in that chapter, the concept of change management is problematic, partly because it could refer to at least three different types of activity. One of these involves the creation of support for the system as it is implemented in its organisational context and the provision of training, time and resources to allow people to effectively utilise the new system. Given the confusion around the term, “change management”, perhaps it would be useful to refer to such activities as “tactics for creating and sustaining commitment to a project and the resulting system”.

Some practical suggestions for creating and sustaining such commitment for a project could include:

- *active* involvement of all who can affect and be affected by the project in the definition and planning stages.
- project marketing and persuasion. Others should be aware of the project and be interested in its proposed outcomes early in the project if their cooperation and involvement is later required. Project marketing is a particularly important issue for whole-of-government projects. The marketing strategy should appreciate differences in separate stakeholder groups and cater for their requirements.
- Project players must be able to legitimise their actions in the eyes of those who are affected or who can affect the project’s outcomes. They need to establish credibility and engender trust. Apart from having demonstrable skills, expertise and experience, ways of legitimising actions include:
 - establishing good personal relationships. Expertise alone does not inspire trust and credibility (Bashein and Markus (1997);
 - illustrating that actions are being driven by the needs of the stakeholders, and that their needs and requirements are being considered seriously;
 - using the recommendations of consultants or established formal methodologies to support outcomes; and
 - involve senior executives as project champions to lend the project authority¹⁸.

¹⁸ These recommendations were included in the Project Management Guidelines released in November 1997. They are currently used by the CIPU and others involved in project management in the State Service.

If a project does not have the necessary support from those providing resources and those who will be utilising the system, it is unlikely to be successful. The creation of a coalition of interest and meaning for the project is important. However, it was not well-achieved in the LSP. Despite these tactics, a significant proportion of the OPC remained uncommitted to the system by the end of the project and the successful implementation of the system ultimately relied on the application of authority and expertise, as discussed in the next chapter.

To recapitulate, system development initiatives need coalitions of commitment if they are to be successful. Systems developers can employ a number of different tactics to promote such commitment, such as marketing and the judicious use of authority, as discussed further in the next chapter. These tactics are important not only for creating a coalition of meaning and interest at the beginning of a project, but for sustaining it throughout and after a project.

3.5 The application of authority and expertise

Despite the utility of the concept of negotiation in explaining social processes surrounding the LSP, sometimes interest and meaning flowed only one way, rather than the two ways suggested by the concept of negotiation. This one way flow involved the application of authority and expertise. These concept can explain significant incidents during the LSP.

If we accept that negotiation is almost ubiquitous during systems development, can and should the process be totally viewed as one of negotiation? If we believe that it can be, then systems development is not a top-down process beginning at the top of the organisational hierarchy as much of the normative systems development literature suggests, but a bottom up one as change is negotiated between all involved.

Yet this ignores that people have differing levels of authority when negotiating change. These aspects of the formal authority structure, as identified in Chapter 2.3, actively shape the process of systems development and direct the outcomes of negotiation. Here it is illustrated that such authority can be derived from hierarchical authority and the application of expertise and the impact of this authority is examined in relation to the LSP.

The concept of negotiation implies a two-way flow of information and power and at times in the LSP there was a one way flow as authority was exercised. Actions were legitimised not because they reflected the outcome of a bargaining process, but because those with authority used their power or expertise to promote it. The concept of negotiation effectively illustrates the interactional nature of information systems development, but tends to downplay issues of domination, power and control. That is, it does not explain why some actors were able to dominate others through the use of power and resources. Sometimes during the LSP meanings and interests simply could not be compromised and conflicts were resolved through the use of authority or the application of expertise.

3.5.1 Is negotiation enough?

The negotiation of interest and meaning is a useful concept, but it does not explain some key aspects of social interaction surrounding the process of planning, developing and implementing systems. It focuses on a two-way flow of power-interactions and meanings, yet there were times in the LSP when these were only one-way flows of either interest or meaning. These one-way flows of information or interests were legitimised in two ways: either the source utilised their recognised expertise and imparted knowledge, or they utilised their position in the organisational

hierarchy to bypass a process of negotiation. The first can be termed the application of expertise; the latter, the use of authority. The application of expertise refers to the dissemination of information and involves learning while the use of authority is the application of power based on one's position in the organisational hierarchy to control others' actions.

The concept of negotiation is based on the assumptions of symbolic interactionism, a stance which Giddens criticises for not adequately dealing with institutional analysis and transformation. The concepts of authority and expertise illustrate the impact of existing structures in the creation of new negotiated order. The rest of this chapter will elaborate on these two processes and illustrate how closely they are related to the concept of negotiation.

Legitimation is a social process by which practices and ideas gain social acceptance (Hirschheim and Klein 1989). Legitimacy can be derived from negotiation and consensus but actions can also be justified through perceived expertise or hierarchical authority. Legitimation is crucial for understanding actions involved in systems development. As Crowe et al (1996) stated,

Legitimacy is rarely the product of unfettered consensus. Rather it is an outcome of the control of resources which render meanings and identities seemingly unproblematical for the practical, historically conditioned purposes of individuals and groups (p 69).

The application of authority and legitimation can also be discussed in terms of power and control of others. These terms have been avoided here due to the loaded meanings, but are commonly used elsewhere. For example, Orlikowski (1988, 1991) focused on control mechanisms in a large accounting/consultancy firm and changes in them when production processes became more mediated by CASE tools. She noted that IT can be both a medium and outcome of social actions, and so can both be influenced by people's actions, and also influence them. These observations are closely aligned with Giddens' structuration theory, and are discussed in more detail in Part 4.

Authority in organisations today tends to be derived from either expertise or hierarchical authority or both. Orlikowski (1988) termed these technical and managerial authority respectively. She noted that, with the growing complexity of production processes and the corresponding greater influence of technical experts and professionals, the tension between the power of knowledge and the power of hierarchical authority has become a major issue. The power of knowledge, "claimed by technical experts and professionals, and accruing to them independent of the organization" can overlap with the power of hierarchical authority, "claimed by organizational managers by virtue of their status in the bureaucratic hierarchy, and not associated with them independently of their office" (p 92). Here these two types of

authority are described in action and the relationship between the two analysed. It will be illustrated that these concepts are integral for understanding key incidents and issues during the LSP.

3.5.2 The application of expertise

The application of expertise refers to a one-way flow of information as one party provides information and the other party learns. As Berger and Luckman (1966) noted, knowledge "...is at the heart of the fundamental dialectic of society. It 'programmes' the channels in which the externalization produces an objective world" (p 84). Pentland (1995) argued that the status of a piece of knowledge depends on the outcomes of debate. As debate converges, issues become settled and take on the character of a "black box". Pichault (1995) cites a number of authors looking at expert power resulting from the possession of specialist skills. He comments that such power distinctions are relatively stable and tend to be an integral part of the formal organisational hierarchy. Therefore, they are difficult to change by simple managerial intervention. This had implication for how the LSP unfolded.

3.5.3 The use of authority

Top management commitment is considered important for system success because they have the authority to promote changes (Cavaye and Cragg 1994). The commercial methodology employed in the LSP commented that it was important to ensure top management commitment as changes are sometimes difficult to achieve "especially in mature and inherently conservative organisations". Managers may simply state they are co-ordinating people and their actions, but, as Wildavsky (1973 in Mintzberg 1994: p 165) pointed out, co-ordination is merely "another term for coercion". The general acceptance that top management support is an important aspect of successful systems development projects provides an acceptable way of expressing what is essentially a taboo issue in organisations. Top management commitment is important because their power and authority legitimises the actions of the systems developers, not just because it helps ensure "the system is aligned with the goals of the organisation".

Reflecting the close relationship between negotiation and the use of authority, Feldman (1989 in Walsham 1993) suggests most management literature is mistaken in viewing control and autonomy as alternatives as they are conceptually and practically inseparable. Autonomy implies autonomy from control and control is an important aspect of managerial actions in organisations. Giddens termed the relationship between control and autonomy the "dialectic of control" and pointed out that pure control of a situation is an impossibility. Even if people did try and follow the rules and directives of those in control, these rules and regulations are always open to

interpretation (Strauss 1987). Negotiation and the use of authority are closely intertwined and crucial elements of change in organisations, such as that involved with the LSP.

3.5.4 Systems developers, authority and expertise

As discussed in Chapter 2.4, systems developers tend to obtain legitimacy for their actions through their technical expertise. Markus and Bjørn-Anderson (1987) suggest that systems developers sometimes used their perceived expertise for leveraging power relations. They looked at the power systems developers had over users and identified four types of power exercise, including:

- the technical exercise of power. Systems developers use their technical expertise to make decisions and do not provide information to others so they could make the decisions themselves. They suggest this occurs often;
- the structural exercise of power by, for example, creating organisational procedures that gives them formal authority over users, or makes users dependent on them;
- the conceptual exercise by, for example, selecting the goals of the system. They point out that many systems development methodologies largely follow the principles of scientific management and so tend to produce high structured jobs and procedures; and
- the symbolic exercise of power, by "shaping users' desire and values outside the context of an individual system development effort" (p 501). Information systems can be embedded with symbols and ideas which may influence people's attitudes about their job and work design. They comment this type of power exercise has been only rarely examined, possibly because both IS professional and users do not acknowledge it. The symbolic aspect of computerisation was examined by Prasad (1993).

Systems development is often likened to learning, emphasising the way that expertise is imparted during the process (eg Crowe, Deeby et al. 1996) but this tends to downplay the influence of power. Conflicts of interest and their resolution tend to be overlooked as such a stance tends to assume that different perspectives can simply be integrated once communicated. In essence, the systems development process involves substantial application of authority and expertise, not only by systems developers, but also line managers, top managers and expert users.

3.5.5 Hierarchical authority versus authority derived from expertise

Orlikowski (1988, 1991) pointed out that power derived from technical expertise can overlap that derived from hierarchical authority and conflict can occur:

As production processes grew increasingly complex, the introduction of technical experts and professionals into the production process grew proportionately. Professionals and technical experts, by their training, socialization, association, and personal interests, remain difficult to integrate into typical bureaucratic organizations... Conflict between the technical experts/ professionals and managers around the production process is not uncommon (1988: p 92).

She predicted that the confusion and confrontation resulting from the overlap of these two types of authority would gradually become resolved as organisations increasingly become dependent on experts and they become embedded in the technical/process infrastructure. Referring to work by Heydebrand (1985), Hirschhorn (1981) and Mintzberg (1979), she predicted this would occur in two possible ways. Firstly, experts could work in multi-person teams with their authority becoming merged with that of functional and line managers and any discrepancies in values, goals and perspectives is resolved within the team. Secondly, experts could become managers themselves, or by managers becoming technical experts. She suggested this synthesis is being promoted by:

- the ongoing push for sophisticated innovation, which requires experts and professionals with the authority to be effective;
- a perceived need by management for co-opting the authority of experts in lieu of being able to control them; and
- the willingness of professional and technical experts to take on hierarchical authority.

These two types of authority can also become integrated by being embedded organisational processes, many of which are an integral part of computerised information systems. These moves reflect the rise of technocratic rationality, as discussed in Chapter 2.4. Expert authority will increasingly become embedded in standard organisational processes, which are governed and policed by those with hierarchical authority. This issue of standardisation in relation to the LSP is addressed further in the following chapter.

3.5.6 The application of expertise and the use of authority in the LSP

While the negotiation of interest and meaning occurred throughout the LSP, the application of expertise and use of authority were also apparent. Here some further incidents which particularly reflect these themes are described and analysed in the light of the theoretical background provided above.

a) Obtaining the active involvement of OPC management

The reaction to the functional requirement specifications and other incidents illustrate that often there were differences of opinion on key issues. While the members of the

OPC agreed to the broad aims and objectives of the project, there were many disagreements over how these objectives should be achieved. However, in order to meet the requirements of the users, some consensus as to what these requirements involved needed to be made within the user organisation. This was an issue in the LSP due to the culture of the OPC and the nature of the changes being considered. Issues arising in the OPC were generally subjected to a great deal of discussion, as intelligent, articulate people debated their advantages, disadvantages, logicity and legality. Conflicting opinions were openly expressed in the office, reflecting its adversarial legal culture. One of the systems developers who had a legal education commented once during discussions, "I'm a student of law. Conflicts of interest are my business" while one member of the OPC warned, "With seven drafters you can easily have ten different opinions" .

The systems developers felt threatened by these confrontations and at times interpreted them as manifestations of conflict within the office. They sometimes felt the users were reacting unreasonably to their work and interpreted the drafters' forceful articulation of opinions as aggressive. They had trouble working out what the requirements of the system would be when there was no consensus of opinion and at one time it seemed the project itself could become a divisive issue in the user organisation. While recognising that the internal operations of the OPC were the concern of the office's management, there were occasions when the lack of consensus within the OPC impinged on what they saw, at least at times, as a technical systems project. The systems developers had to cope with the internal debates surrounding this situation, as they attempted to determine areas of consensus in key areas and gain the commitment to the system and the project.

As discussed previously, the chief drafter was considered to be an excellent drafter and tended to delegate a large proportion of the management of the office to his deputy due to his and the office's heavy drafting workload. The chief drafter had delegated responsibility for the LSP to the deputy manager and the systems developers' contact with most other people in the office was limited and sometimes actively discouraged. Others complained they knew very little about the project, and the LSP team were concerned their commitment to the project and the system would not be achieved. The deputy manager had been very helpful in explaining the process of drafting but the reaction of others to the functional requirements document illustrated the use of different methods and a lack of communication within the office. Individual members of the OPC expressed a variety of different opinions and the systems developers had trouble finding firm requirements on which to build the technical system.

They raised this issue with the chief drafter, who subsequently took on a more active role in the project, despite his ongoing heavy drafting commitments. The systems developers continually emphasised the chief drafter's responsibilities and how he would make any final decisions in the office. Aware that his active involvement was required in the project, the chief drafter came to play a crucial role as he negotiated or enforced decisions on key issues. While previously the chief drafter had let others run office meetings concerning the LSP, he now generally sat at the head of the table and at least introduced them and, whereas before he tended to concentrate on his drafting workload in preference to attending all LSP meetings, he now took a large role in discussions within them. The chief drafter tried to incorporate the different views in the office, while not letting the diversity of perceptions and opinions interfere in the effective workings of the office and the LSP.

The chief drafter had obtained his position because he was a senior drafter with recognised expertise in creating complex pieces of legislation. His ability as a drafter gave him considerable respect and it is unlikely that the drafters would have accepted a non-drafter as a manager. They often emphasised that non-drafters did not appreciate the challenges they faced and a suggestion to reduce some of the OPC management's heavy workload by employing an office manager was not pursued. Expert and hierarchical authority was combined in the chief drafter's position.

The chief drafter was very much a practising drafter as well as a manager. He would specifically be asked to create sophisticated, often politically sensitive and urgent draft legislation by members of Cabinet. Combining heavy drafting commitments with involvement in the LSP and other office affairs proved an ongoing challenge. Although he and others hoped for his ongoing active involvement, in many cases he had to delegate decision making to others. However, he and the deputy chief drafter would review the outcomes and take on responsibility for them, so exercising managerial or hierarchical authority.

The OPC is essentially an organisation of professionals. All the drafters, and at least some of the support staff to a degree, had power due to their professional or technical expertise. It could be a very challenging office to manage, as this power derived from expertise at times confronted the authority derived from hierarchical positioning.

However, although others in the office would have had the technical expertise to make many of these decisions, they generally did not have the authority to enforce them. Thus, the OPC management's "stamp of approval" was crucial if the decisions were to be implemented. As the head of the OPC, the chief drafter had personal responsibility for activities within the office. The other members of the OPC generally appreciated that, if he had responsibility for the decisions, the chief drafter

should be comfortable with those decisions. Thus, despite their own personal competency and often different opinions, they generally followed his directives.

At the same time, the expertise of others in the office was also acknowledged by delegating decision-making to them. OPC staff members would work through the details required to make a specific decision. They would make recommendations to OPC management which were, generally, accepted. Hence, expertise and hierarchical authority were combined in the OPC through having experienced drafters as managers and by delegating decision making to specific individuals. Hierarchical and expert authority proved challenging forces to balance in the office, both generally and in terms of the social interaction surrounding the LSP.

b) Development of proposed technical solution by user

One area of debate revolved around the drafting environment the new system would provide. Many of the drafters were concerned the analysis and prototypes did not accurately reflect their work practices, particularly the assumption the drafters wrote in segments. They generally described how drafting took a great deal of concentration and they were concerned the new tools could interfere with this.

One senior drafter had enough experience and expertise in computing as well as the creativity and interest to develop a prototype drafting environment for drafting amendments. He developed it in response to the inadequacies perceived in the prototypes and functional specification and frustration resulting from the belief the systems developers were not making any progress in the development of the system.

I was away when the [first] prototype... was given a trial in the Office, but from all accounts it was quite inappropriate. I believe that the system was incorrectly based on a theory of segments. On the other hand, Word 5 in a Windows for Workgroup environment would be ideal for our purposes (Comments on Functional Requirements Specification 5/7/94).

We must get away from this going around in circles action. Everything is waiting for something else to happen... we need to start doing something or this whole thing will be an Alice in Wonderland exercise (transcripts 26/8/94).

He developed the prototype using the technology he had most experience with and had the sort of user interface he believed such a system should have. Some of the other drafters agreed this prototype reflected their work practices much better than the prototypes presented by the systems analyst.

I'm starting to get confident with the LSP after talking with [the drafter who developed the prototype] I think I could use this and not only that, to my advantage. It's pretty good. It seems to be fully in place. It is good it actually comes from a drafter (transcripts 16/8/94).

...it works in the way we work (transcripts 6/9/94).

The drafter who had developed the prototype believed that he had saved the LSP team considerable effort by developing part of a technical system which would meet the needs of the drafters. By combining his extensive drafting expertise with his limited

experience in computers, he was able to make what he saw a valuable contribution to the project.

The technology used for this prototype was relatively old and did not include some of the features the LSP team believed should be in the new system, such as pull-down menus and mouse-driven controls. The developers were also concerned that, while the system might meet the requirements of the drafters, it might not meet those of others. Additionally, the technical group had experience with a previous systems development project which had suffered from a lack of planning and had been strongly influenced by the IT management planning literature which emphasised the need to carefully plan information systems development projects.

The drafter who had developed the prototype could not understand why the system he had produced was not being adopted by the systems developers. This person had both drafting and computing skills, but others saw his suggestions being dismissed by the systems developers. While stating they wished to listen to the users' viewpoints and wanted their input, the LSP team would not listen when one of them expressed firm ideas. The systems developers spent considerable time looking at the user-developed prototype, even though they had dismissed it almost immediately, wishing to illustrate that they were interested in what users had to say. They explained these reasons to the managers of the drafting organisation who passed this explanation on to the rest of the OPC. This was an awkward situation for all involved.

The systems developers had to respond to two broad issues. The first focuses on the suitability of the technology proposed by the user to meet the demands of the full system while the second concerns the conceptual structure of the proposed drafting environment. In reality, these two issues became intermingled in the perceptions and actions of those involved, but logically they are separate and resulted in different outcomes.

The conceptual structure of the drafting process was an ongoing issue. There were numerous discussions concerning the belief by the systems developers that the drafters wrote in sections, an assumption which had been strongly reflected in both the system developers' prototypes and the functional requirements document. The user developed prototype presented a different scenario. Trying to fit the new system into the drafters' existing working practices, they had based their conceptual structure of the drafting environment on the formal method of inserting amendments. Formally, amendments were written using this "insert/omit" method¹⁹, but often in practice they initially created amendments, even just in their minds, by an implicit

¹⁹ See Appendix 4 for an explanation of these two different approaches.

strikethrough/underline approach before translating it into the formal insert/omit documents. That is, generally they would develop their ideas concerning the appropriate amendments in relation to the principal bill before producing the separate amending document. In the early stages of the project, the feasibility of changing amendments to the strike through/underline style had been discussed but was rejected. This possibly blinded the LSP team to the way the drafters actually wrote amendments.

Unfortunately, in the eyes of those involved, this conceptual issue became tied up with the technical solution proposed. The user continually brought up the conceptual issue and presented his technical system as a solution. The systems developers tended to dismiss the technical solution due to the outdated technology but, at least initially, overlooked the conceptual issues being raised. They became annoyed that the drafter kept strongly promoting the technology, but seemed unable to look beyond the outdated technology to the conceptual information the drafter was providing. The conceptual approach the drafter used in his prototype was eventually embedded in later versions of the functional requirements document but some of the drafters felt for quite some time that the CIPU did not understand their way of working.

The systems developers attempted to convince the other members of the OPC the prototype was technologically inadequate by recruiting the technical expertise of consultants. They organised a meeting in which the drafters could discuss how amendment operations could technically be achieved and invited a person from one of the tendering companies with extensive expertise in desktop publishing to participate (6/9/94). The drafters were invited to ask this person about technical issues, such as methods of amending legislation by computer. The business analyst found this a very trying exercise as the drafter who had developed the prototype emphasised that the whole exercise was a waste of time as his prototype had already solved the problem. During the meeting the drafter commented,

There is no need for any of this. It is all set up on [one of the administrative assistant's machines]. All we need is an over-riding macro. With my system, if you select the text and hit "O", you can omit a word, so we don't need templates or any of this. That's all you need and it's all set up (transcripts 6/9/94).

While later one of the LSP team members wrote,

[The drafter who developed the prototype] is monopolising group discussions which is preventing the drafters advising the CIPU of their requirements (LSP project documentation 6/9/94).

During the meeting the business analyst emphasised the need to carefully examine the process of producing amendments and the requirements of everyone. She suggested that being able to use a mouse might be useful to which the drafter responded he could not see the purpose of creating an entirely different system just to incorporate mouse controls. Another drafter added that the administrative assistants had found

the mouse useful. The tendering consultant suggested there were also significant advantages to using pull-down menus while the drafter who had developed the prototype commented a control key could be used to see the menu with his system. The tendering consultant suggested that, with the development of GUI interfaces, the days of such menus were over, commenting individuals in the office could even design their own menus, and added,

There is resistance to some ideas from my perspective (6/9/94).

The business analyst was concerned by this incident and the project director summed up an interpretation of the situation for her in jest:

You're there, stuffing around, not producing anything and then one of them comes up with something that looks like it does the job and they want to roll with that. (The business analyst groans.)

But this is not a totally negative situation. This is the first time the drafters have actually really related to this [project] in a constructive way... you may look like an idiot, but you are getting them onside to the project (17/8/94).

At this time, the systems developers were spending a great deal of time developing an appropriate tender evaluation scheme and the user who had developed the prototype system could not understand why all this effort was being expended when he had already developed a system which worked technically and which in his opinion reflected the working requirements of the drafters. He was keen to show the prototype to anyone who might support him, including myself. At this stage, it was clear if I showed interest, the drafter could be encouraged to continue promoting his prototype. I maintained that I did not have enough expertise to adequately judge it and had no influence anyway and did not actively pursue the topic. This situation illustrates the difficulty in remaining a purely passive observer in a social situation. It also effectively illustrated that conflict can occur between hierarchical and expert authority, but also between different types of expert authority. This incident reveals a collision of competing areas of expertise which was resolved, at least superficially, through the application of hierarchical authority.

The systems developers repeated the user-developed prototype used old technology and emphasised the need to carefully plan the new information system and consider all possible options before the system was implemented. They explained this to the steering committee, the management of the OPC and to the user several times. The user was asked to drop the issue by user management. I am aware of no further discussions on the issue although it remained a concern for the LSP team throughout the rest of the project.

This incident reflects the interconnected nature of negotiation, authority and expertise particularly well. The prototype developed by the user provided a forum of negotiation but the issue was resolved with the aid of expertise and authority. By

combining the drafter's drafting expertise with his more limited knowledge of computerised technology, the prototype provided something by which to judge the solution envisaged by the LSP team. When negotiations failed to resolve the issue, the systems developer's expertise combined with the authority of the managers involved delegitimised the actions of the drafter.

c) Use of Consultants

The CIPU commonly used external consultants and contractors to both supplement the five or six people who worked in the unit on a full-time basis and to have access to specialised knowledge. The difference between contractors and consultants is that the former performed some tasks on behalf of the CIPU while consultants merely provided advice. Some contractors only provided extra labour in specific areas (such as data conversion or project management) but many contributed specific skills. For instance, the contractor employed to develop the word processing templates and camera ready processes had recognised skills in word processing technologies.

Examples of the types of consultants used during the LSP include information technology specialists, project management and quality review experts, legal advisers, for the review and development of contractual documents, documentation, training and change management specialists. Their influence illustrates the impact of expertise on the project. For example, the CIPU asked project management consultants to review their progress regularly and provide some input into current issues of concern. These consultants also provided a periodic report to the steering committee and so acted an auditing role in the project. Later, as the project management methodology employed by the CIPU matured, these consultants were redefined as Quality Review consultants. Formally, their task was to ensure the project management activities were appropriate for the project and assist the project's steering committee achieve the project's outcomes.

The roles of consultants in providing expertise is a particularly pertinent role in systems development. Yet is it only expertise that they are providing? Certainly the consultants in the LSP often provided information or services the members of the CIPU were simply not able to, but they also helped to *externalise* or *rationalise* the actions and beliefs of the internal systems developers, giving them a sense of objectivity, or rationality. In other words, their actions were being justified by outsiders. In at least some cases, observations made by external consultants would not have provided the systems developers with more information, but they did articulate what internal people were saying, though less clearly in relation to issues such as the involvement of users. In addition to formally articulating the beliefs of the internal systems developers and so contributing to the on-going process of sense-

making, the consultants also helped to rationalise the actions of systems developers by spreading responsibility for them to external parties.

These observations echo those of Pfeffer (1981 in Lacity and Hirschheim 1993). As well as providing expert knowledge, such consultants can provide information about what is happening in other organisations and the fact they seen as external can provide a sense of objectivity to their recommendations. As Pfeffer noted, outside consultants rarely provide novel recommendations but tend to formalise their clients' predetermined intentions, so providing legitimacy to decisions. Consultants can have a large influence on projects such as the LSP.

d) Controlling the achievement of objectives: outputs and outcomes

In the latter part of the project, the CIPU employed a new project management consultant to review their progress on an ongoing basis. This consultant contributed some new theories to aid their actions. One of these was the distinction between the outputs and outcomes of a project. A project's outputs include specific changes, such as the implementation of a new computer system, or new procedures. These outputs are not beneficial in themselves, but they do enable outcomes, which are. These outcomes are linked with organisational goals.

The primary difference between an output and an outcome, as far as the CIPU were concerned, was that outputs were seen as controllable, whereas outcomes were not. The proposed outcomes of the project, which included the benefits to be derived from the OPC actively and efficiently using the system, were uncontrollable and thus not within the scope of the CIPU's arena of management responsibility. In other words, the CIPU could expect to have power, or authority, over outputs, but not necessarily the achievement of outcomes.

Although this model has its benefits, its application in the short term in this situation led to the members of the CIPU simply abrogating responsibility in some areas. Asked about the implementation of the EnAct system in the OPC in late 1996, the project manager said this was not the concern of the CIPU team.

This issue illustrates the theme of the application of authority and expertise in a number of ways. Firstly, it illustrates how the expertise of consultants can influence the process of system development and perceived areas of authority. Particularly, it illustrates control over others' actions and future events, an issue pursued further in Part 4. The distinction between outputs and outcomes is an attempt to separate what can be managed and controlled from what cannot. The distinction illustrated the importance that systems developers place on controlling events. It allowed them to ignore messy social issues and remain focused on what they could control for a time.

This move represented a retreat into rationality as issues of involvement and user acceptance could be conceptually separated from a systems development project.

Thankfully, the project management consultant and the CIPU seemed to become aware of the potential problems associated with not promoting the outcomes of the project. The project management consultant suggested:

...At the moment, the LSP project is dominated by a systems deliverable - the [EnAct] application suite. There appears to be a general expectation that, when delivered to the OPC, [EnAct] will be used selectively by staff (after some training). This expectation is based on the assumption that existing processes within the OPC remain largely unchanged by the project (Project management consultant's report 23/7/1997).

It is common to find that, because of the peculiar pressures of implementation, teams tend to substitute a goal of "get the system in" for the original objective of "generate the target benefits" (Project management consultant report 24/10/1997).

As the project came closer to being technically complete, the CIPU realised there needed to be at least the active consideration of procedural and other organisational changes if the technical project was to be a success. The OPC found it difficult to comprehend, let alone plan for such changes when most of them had not seen the system in action.

3.5.7 Reflections: hierarchical and expert authority

Until this chapter, the focus on Part 3 has been on the active role that participants play in information systems development projects and how the process can be viewed as one of negotiation between these participants. However, the negotiation playing field was not always level. The LSP would not have progressed without key participants employing the authority their positions offered them. The executive manager responsible for both the OPC and the CIPU heavily promoted the project and convinced Cabinet of its benefits. The support of Cabinet and other bodies was used to obtain the commitment of the OPC while the management of the OPC made decisions which the other members of the office were obliged to follow. The project leader made decisions which the members of his team implemented. The drafter who developed the prototype had been asked to drop the issue, the analyst and consultant followed the directives of the project leader and the steering committee kept tabs on what outcomes the project would produce and how.

Through the use of power and authority, people were able to legitimise the actions of individuals by giving the support of the organisational power hierarchies. With the weight of authority, decisions were made objective facts in the eyes of those who had to follow them, rather than a negotiated reality which could be debated. Some other examples of the use of authority and expertise from other times in the project are outlined in Table 3.5.1.

date	incident	interpretation
21/7/94 meeting with admin staff by one of LSP team	"This group has voiced their concern about the changes under the LSP- ie. they don't want to be in a position of <i>checking</i> drafts according to office guidelines unless they receive external support (preferably external to the office)."	The administrative staff do not feel the drafters will recognise they have the authority to make decisions which may conflict with their actions. In other words, the hierarchical authority of the administrative assistance (having been given the authority to enforce standards on the drafters) may conflict with the expertise of the drafters and the drafters would not appreciate this. Thus the systems developers acknowledge that the authority of the administrative assistants would have to be reinforced.
9/8/94 meeting with OPC manage ment by LSP team member	"[The deputy chief drafter] has concerns that CIPU staff are taking suggestions from other staff on face value and not raising them with OPC management. [LSP team member] assured [the deputy manager] that she was not acting on any suggestion without [the chief drafter's] approval and <i>would</i> raise any issues that she felt needed pursuing or bringing to [OPC management's] attention."	The OPC manager believes that their hierarchical authority is being eroded by the systems developers' actions and reinforces their authority by requiring that all decisions be at least okayed by them.
4/11/94	Users told LSP team needed to get endorsement for proof of concept stage from steering committee	The steering committee has the authority to make the decision.
13/9/95	Parliamentary government forces OPC to implement camera ready statutory rules part-way through a parliamentary session	OPC to follow directives of Parliamentary executive government even though timing was inconvenient.
1/11/95 project doc'n	Chief drafter required for detailed design and acceptance testing ostensibly for drafting expertise	The drafting expertise of all or any of the drafters was required during these stages. Why was the chief drafter picked out? He would provide the authority for any decisions made. Not only would decisions made have legitimacy of being okayed by someone with drafting expertise, but they would also be legitimised by someone with formal hierarchical authority.

Table 3.5.1: Examples of the application of authority and expertise during the early stages of the LSP

However, power did not just flow in one direction. The proposed outcomes of the project were developed through discussions between all involved during incidents such as the prototype evaluation, the review of the functional requirements specification and so forth. Before asking the drafter who developed the prototype to drop the issue, the chief drafter listened to what he proposed and broadly supported the conceptual aspect of his program. Generally, the chief drafter only made decisions concerning the drafting process when no consensus could be reached or the decision reached through negotiations conflicted with external requirements of the office or the project. The steering committee made its decisions based on the recommendations of those briefing them, as did Cabinet.

Although the development of the LSP involved a process of mutual learning as systems developers, members of the OPC and others learnt about the worldviews of the others, there were significant times when a one-way flow of information could be observed. Such one-way flows of information included:

- training programs. These included wordprocessing refresher training courses for the administrative assistants, a computer awareness course and training sessions for the system itself, as described in other chapters;
- a trip to another state's OPC to look at their application of computerised technology by one of the systems developers and one of the drafters;
- documentation for new people in the project so they could understand the details of the project and its context; and
- expert advice was provided through consultants. Such interactions were primarily a one-way flow of information (after the provider had ascertained what the recipient wanted or needed to know).

The one way flow of information can serve as a tool for the two-way process of negotiating interests. For example:

- one of the aims of the prototype was "to illustrate to the chief drafter how quick and easy it is";
- systems developers say the users need to be "educated" about the benefits of the system; and
- administrative assistants say the drafters need to be "educated" about how difficult the administrative assistants found it to keep up with their demands.

The process of systems development in projects such as the LSP is essentially an interaction between people with different areas of expertise. The OPC staff members knew a great deal about the process of producing legislation but generally knew very little about computerised technology at the beginning of the project. The systems developers had to learn a great deal about the production of legislation to apply their area of expertise. The negotiation of meaning often involved the transferral of information from one group to another. This one-way flow of information was legitimised by the fact that the providers were seen to have expertise in that area.

The authority derived from expertise was enhanced by and sometimes conflicted with authority derived from hierarchical structures. The executive manager who initiated and promoted the project and the commitment of the parliamentary leaders were crucial because they helped to authorise and legitimise the actions of the systems developers. Throughout the project, the executive manager stated that it was a

requirement of Cabinet that the project was implemented and when consensus could not be achieved in the OPC, the chief drafter made a decision the others generally followed. Although often viewed with negative connotations, issues of control and the use of authority can be viewed positively and are one way of resolving or circumventing conflicts of interest and meaning and/or speeding up the process of negotiation.

As Markus and Bjørn-Anderson (1987) pointed out, systems developers can exercise power in a number of ways, and this was done in the LSP. For example, via

- the technical exercise of power. The CIPU and the technical systems developers often made technical decisions on behalf of the OPC, who had limited computing knowledge. For example, while allowing the OPC to choose which SGML editor was suitable for their purposes, they could only choose from a short-list created by the technical systems contractors based on technical and cost considerations;
- the structural exercise of power. The formal systems development and project management methodologies used by the CIPU prescribed the roles and responsibilities of the OPC. For example, they had to “sign off” the functional requirements document and were then responsible for any future changes identified;
- the conceptual exercise of power. The information system proposed through the LSP firmly reflected the technocratic assumptions of the CIPU, as illustrated in the following chapter. As with most information systems, the implementation of EnAct promoted standardisation of procedures; and
- the symbolic exercise of power. This was not specifically examined in this research project and can be difficult to reveal.

The drafters also used their expertise to emphasise particular points and the specialised and valuable nature of their expertise gave them a particularly good bargaining tool. The CIPU acknowledged that the acceptance and effective utilisation of the EnAct system was crucial and a considerable risk throughout the project. The systems development process thus involved the application of expertise in a number of ways.

Negotiation can be defined as the two-way flow of meaning and interest (power) between people involved in systems development. The use of authority refers to a one-way flow of interest, or power, while the application of expertise is the one-way flow of information, or meaning. These concepts are closely intertwined and to dismiss one would ignore important facets of the social interactions surrounding systems development. Without negotiation, there is a risk that some parties are relegated to passive involvement. Yet if we ignore the use of expertise or authority,

we forget that some people have more influence over the situation than others due to the information they impart or the power they can leverage.

The emphasis on negotiation here is in response to an over-concentration on the use of authority and expertise in much of the normative literature. In this literature, the prerogative of management tends to be over-emphasised and conflicts of interest and meaning relegated to "resistance" to be "managed" or a matter of training and education. Overcoming these problems has come to be termed "change management" and this is pursued in the chapter after next.

3.6 Reflections: The LSP and change in the OPC

To complete this examination of how the content of the LSP was established, this final chapter of Part 3 will outline what changes have occurred to date. While many of the implications of the LSP are still to be felt, the EnAct system has already promoted significant changes, particularly in the OPC. Outside the OPC, there have not been, and are unlikely to be, substantial direct impacts on the enactment of legislation, though it does have a huge impact on people's access to legislation and could have significant longer-term indirect effects. Within the OPC, the EnAct system seems to be changing the way that drafters create legislation and has induced standardisation. Other role, authority and cultural changes are still unfolding, but could be significant in the longer term. These changes occurred as a result of the social, or political, processes which make up systems development at a micro level, and involved active participation, the negotiation of meaning and interest, the creation and sustainment of coalitions commitment and the application of authority and expertise, as covered in the previous chapters.

So what changes have occurred as a result of the LSP apart from the implementation of the actual EnAct system? This cannot be answered definitively for a number of reasons, the most obvious being that the system was implemented just as this dissertation was being completed, after a very long delay. Additionally, changes associated with the implementation of new technological systems do not simply occur the day the new system is implemented, but take a while to unfurl, as people both adapt their work practices to the new technology and use the technology to suit their work practices. This is an ongoing process, which this research project can only touch upon. Some of these changes can be anticipated, but could be the focus of a whole new research project. As Giddens (1979) commented, the results of any action can result in unanticipated consequences. Not all the consequences of change projects such as the LSP can be predicted, just as not all predicted changes are likely to occur (Orlikowski and Hofman 1997).

The fact that the system was implemented only in the closing stages of this research project is only one reason why it difficult to ascertain what these changes precisely involve. Additionally, it is difficult to firmly claim direct correspondence between changes and their results in a real world situation where there are so many other factors which could have influenced the OPC. Other parallel changes, such as the implementation of enterprise bargaining, plain English drafting, changes in the executive government after the election in early 1996 and broad social changes may have contributed to the changes observed in and around the OPC. For example,

during the time EnAct was being developed and implemented, OPC staff members became more technically literate. This is probably largely due to the training programs held in conjunction with the project, but it may also have been affected by a growing general awareness of technology and technological issues by the general public. Furthermore, cultural changes are difficult to ascertain because they exist in the minds and interactions of people involved.

Some changes in the OPC seemed to be linked with the process of developing the LSP rather than with the implementation of the EnAct system. That is, the process of developing the system has produced changes, even before the content of the project was implemented. The project and the involvement of other people in the OPC seemed to stimulate change before the system was implemented, and many of these changes were not envisaged or planned.

Nevertheless, many changes were observable even before the full implementation of the system and some possible contextual changes can be anticipated. The changes observed are described below in terms of the relevant contextual issues discussed in Part 2.

Orlikowski (1988) predicted that, as organisations adopt information technology as their production technology to perform their business operations, associated changes would affect:

- the nature of production tasks, such as:
 - a shift towards technical rationality;
 - increased technical and cultural control;
 - deskilling of production tasks;
 - increase of formalisation, abstraction and reification of task content;
 - integration of the division of labour.
- the expertise associated with the production processes:
 - a synthesis of technical expert and managerial roles, achieved through joint project teams or technical managers;
 - a conflict over functional and technical territorialism.
- the production strategy underlying the production process, such as:
 - a shift towards generalised problem-solving approaches;
 - increased programmed customisation;
- organisation of people around the production processes:

- increased debureaucratisation, with more client participation, multi-disciplinary project teams and loosely coupled semi-autonomous production units;
- decentralisation supported with elements of centralisation, as control is centralised (Orlikowski 1988: p 110).

Orlikowski illustrated the manner in which these predictions unfolded in relation to the implementation of CASE tools in a large accounting/ consulting firm. The discussion below will examine the degree to which these predictions seem to be occurring in relation to the OPC as a result of the LSP and the EnAct system. This analysis is limited due to the reasons outlined above, but provide an insight into some of the non-technical changes associated with the implementation of the EnAct system. It will illustrate that many of these consequences of actions surrounding the LSP were at least partly unplanned and/or unforeseen and could not be controlled or managed.

3.6.1 Purpose of the organisation and place in wider context

A parliamentary system created now would probably be quite different to ones created last century because the technology on which the system could be based would be quite different. Procedures for producing legislation that utilise computerised information technology could theoretically be quite different from ones that do not. Yet would the implementation of a computerised information system greatly affect existing procedures for producing legislation?

There has been a growing awareness in the information technology (IT) literature on the organisational change issues surrounding the implementation of information technology. Such literature has focused on both the implications of organisational, or non-technical issues on the development and implementation of technical systems, but also on the organisational changes resulting from the implementation of information technology. In much the same way that cars were once considered horseless carriages, information technology has often been used to simply automate existing processes and work roles, with word processors being used as glorified typewriters. There has been a recognition that such an approach does not make effective use of the technology and so some IT literature has come to focus on the effective utilisation of technologies by transforming organisations (Ward, Griffiths et al. 1990).

This trend has perhaps peaked in literature on business process re-engineering (BPR). The aim of writers in this area, such as Hammer and Davenport (Davenport and Short 1990; Hammer 1990; Hammer and Champy 1990; Davenport and Stoddard 1994) ,

has been to re-evaluate existing organisational processes and to redesign them utilising advanced information technologies to improve the effectiveness and efficiency of these processes. A BPR exercise focusing on the process of producing legislation *a la* Hammer or Davenport would try to redesign the process without considering the roles played by various bodies, the focus being on producing legislation effectively. Such an approach may be a useful way for considering improvements to enhance the efficiency and effectiveness of many organisational processes, but its application in the process of producing legislation flounders on two issues addressed in depth later, but introduced here.

The first, the implementability of such changes, is an issue to which much BPR literature has been turning recently. As Davenport and Stoddard (1994) has acknowledged, a 'blank sheet of paper' approach to designing change usually relies on a 'blank cheque' for implementation. Yet still much of the literature in the area defines problems associated with BPR as implementation issues, rather than as a problem with BPR itself (Craig and Yetton 1992). Hammer, perhaps the best-known writer in this area, has acknowledged that as many as 70% of BPR projects fail due to implementation problems, such as "a lack of top management commitment".

This aligns with much of the literature in strategic planning and policy planning, which views the implementation of changes involved as a separate activity governed by the plans or policies defined earlier. As will be discussed in depth in Part 4, such activities are not necessarily sequential, and involve the management of emergent as well as planned factors. Craig and Yetton (1992; 1994) make a similar point when reviewing the literature on BPR by suggesting that it is through the implementation of BPR that strategic options will emerge; that is, such activities unfold in an emergent fashion. A emergent approach to change is recommended for two main reasons. The first is that change generally occurs in a dynamic context.

The second reason for a contingent approach to change is also the second reason why the application of BPR in the process of producing legislation would probably falter. Change can involve alterations in power relationships in and between organisations and the planning of such changes implicitly involves the attainment of some kind of consensus (or cohesion) as to what these changes involve. Debates surrounding the process of producing legislation are more likely to become issues for formal political debate in parliament and in the community, even though most of the changes potentially introduced by the LSP would directly affect the technical or administrative processes rather than the legislative debating. In the same way that a particular voting system can determine who is elected to parliament, the administrative or technical procedures within Parliament can determine how policies are debated. If some groups believe their interests are not being served by changes to the administrative or

technical procedures in Parliament, those changes will become controversial and hotly debated.

These debates could effectively stall the technical systems project and it would be judged a failure. Most organisational change efforts are concerned with the achievement of consensus, yet technological change requires such consensus in order to begin (Brinkman 1991). In other words, such debates are likely to adversely affect the outcome of any related information systems development project according to the measures of success used in this area. The detailed design process and programming cannot commence without a reasonable degree of certainty as to the goals and broad content of the changes. The use of prototyping can be used in such a situation, but generally increases the resources and time required for the project. This issue of producing a consensus on which to build a system was discussed in Part 3.

There have been predictions that information technology will greatly change the internal workings of government. Brussard (1988) maintained that the structure of a government's public administration is implicitly dependent on the information technologies available. Therefore, information technologies will influence public sector organisations and their relationship with society as a whole. However, the executive branch of government is lagging behind almost all other social institutions in installing new technologies (Abrahamson 1991). Abrahamson suggests this is because the implementation of new technologies can subtly alter the balance of power between government players and the subsequent debates stifle the development of technological systems.

The process of producing legislation is broadly defined in parliamentary standing orders and is embedded in parliamentary and bureaucratic institutions. As one of the consultants involved in the LSP commented,

An additional constraint is that presentation of Bills and amendments must conform to Parliamentary Standing Orders (Project Management Review 8/12/93).

Business process reengineering (BPR) projects aim to re-examine the role of institutions in such processes (Davenport and Short 1990), but in this case it is not a practical option to implement many of the changes such an examination would produce. Parliamentary processes aim not so much for efficiency, as is the focus of BPR activities, but the amalgamation of different opinions and interests into policies. Any alteration to workings of parliament especially could be interpreted as a threat to the existing balance of power between the different political groups in parliament and the resulting debate would probably be significant and lengthy. Any BPR exercise is likely to challenge existing power bases but, in this case, the issues are magnified in that the status quo is the formal system of government. Hence, the people involved in

the LSP have necessarily avoided changing or challenging the workings of parliament.

This strategy reflects Kraemer's (1991) argument that information systems do not induce reform in organisations, but tend to reinforce existing organisational arrangements and power distributions and Brinckman's belief that advanced information technology is rarely used for organisational innovation (Brinckman 1991; Kraemer 1991). Those responsible for the LSP seem to implicitly reflect Kraemer's assertion:

Although computing can in theory lead to new administrative structures, in practice, it doesn't, it can't, and it probably shouldn't. And each time such structures have been attempted in the past disaster has resulted. The disasters will continue as long as the role of information technology in administrative reform is viewed from the perspective of management rationalism and the structural and behavioral realities of organizational power and politics are ignored (Kraemer 1991: p 178)

The discussion above does not dispute that information technology can have an effect on organisational processes. It merely maintains that the difficulty of implementing such changes will slow these trends and that large scale organisational changes will not occur as a result of single projects such as the LSP. The LSP may induce future major change projects outside the OPC but will probably not incur them directly as such changes would be very difficult to manage at the same time.

There have been many predictions that computerised information systems will have a great impact on the workings of government in general (Kraemer 1991).

Theoretically, politicians could sit in front of terminals or workstations in parliament with access to electronic versions of principal bills and amendments in either consolidated or unconsolidated forms. Drafters could be involved in real-time amendments to these bills and amendments and this role of the government printer could be eliminated as drafters type the legislation directly onto the system the members of parliament access.

However, as discussed above, in the short term it is unlikely to occur as a direct impact of a single systems development project. As Chartrand and Ketcham (1994) stated:

In writing a report about information resources and technology for Congress, it is tempting to make the statement that Congress will conduct its business in a vastly different way 20 years from now.

But such a conclusion does not hold up well, if past performance is reflected in present reality. Congress by its nature, as the representative branch of our constitutional government, is cautious about change; it views the prospect of altering the manner in which it performs its day-to-day tasks with some scepticism (p182).

The reaction of readers to the paragraph above this discussing possibilities for the computerisation of parliament is illustrative, with a senior drafter stating he did "not understand this". If systems development occurs as a result of negotiation, they are unlikely to occur immediately or in "one hit".

Chartrand and Ketcham's (1994) observations support Snellen and Schokker's (1992) conclusion that information technology developments within public administration are more complex as a result of the inevitable interweaving of political, judicial and technical aspects. The developers of the LSP coped with this increased complexity by avoiding it. In other words, they avoided having to deal with the broad contextual issues and cope with the corresponding increasing risks by embedding the project in the outer organisational context. Thus, information technology has been predicted to induce great organisational changes but, while the LSP will promote changes in and around the OPC, changes directly resulting from the system on the broader parliamentary procedures will probably be minimal. However, the project was predicted to greatly impact on the legislation drafters and their support staff, and the bulk of the description here focuses on this level of analysis.

Broadly speaking, the purpose of the OPC and its place in its wider organisational context have not changed during the time the LSP was planned, developed and implemented. Perhaps the most significant change in this area was that the OPC took over responsibility for producing camera-ready versions of legislation from the government printing office. The development of the EnAct system resulting from the LSP largely had to fit in with existing processes outside the OPC. These existing processes included the development of parliamentary amendments and the printing of legislation by the government printing office, as well as the broader processes for generally producing legislation.

The aim of the project was to improve the efficiency of the legislation production process and provide facilities for access to automatically consolidated legislation. It is too early to judge if the efficiency goals have been met. The project provided a database of consolidated legislation and facilities for automatic consolidation for future acts and amendments. Access to this database for the general public and other government bodies apart from the OPC was redefined as a second stage of the project, in order to meet budgetary and other constraints on the project. This Legislative Publication Project (LPP) is currently underway.

Other changes in the wider context associated with the LSP include:

- *changes to relevant pieces of legislation.* When the EnAct system was implemented, legislation was stored in a consolidated, electronic format and there were concerns that the legality or evidentiary status of this legislation could be questioned. The solicitor general initially commented:

In order for the electronic version of Tasmanian legislation to acquire legal standing, legislation will certainly be required. The only place where a dispute as to what is or what is not law can be resolved is in the courts, and it is only through legislation that an Act of Parliament not currently in a legally admissible form can be given appropriate evidentiary status...

It appears to me there are two concerns central to the new system. First, there is the question of the legal status which the electronic record is to enjoy, and secondly there is the question of the evidentiary status which it is to have.

In relation to the first question, I imagine that the source document will still be the original hard copy Act filed in the Supreme Court...(11/12/95).

Thus, initially the Solicitor General attempted to fit the changes to the established norms of keeping official versions of documents on paper.

However, evidentiary issues also had to be addressed in relation to electronic copies of legislation. If people were to obtain access to authorised versions of legislation via electronic means, these electronic versions had to be legally recognised as the official source of the law. Before the EnAct system was formally launched, it was decided that electronic copies of legislation should:

- be given at least equivalent status to printed copies; and
- prima facie be evidence of the written law as at a given date.

These issues entailed required changes to the existing *Acts Reprinting Act*. and the *Evidence Act*. The *Evidence Act* was adapted so evidentiary status was given to electronic copies of legislation. The *Acts Reprinting Act* gave the OPC authority to produce consolidated versions of acts and correct minor errors such as some spelling or grammar mistakes. However, it did not cover automatic consolidation. The *Acts Reprinting Act* was superseded by the *Legislation Publication Act 1996*, passed by Parliament in June 1996. This act legally established an authorised database of Tasmanian legislation in electronic form and gave evidentiary status to copies of legislation taken from the database. It also extended the OPC's power to make editorial changes to legislation.

- *changes in the process of forwarding legislation to the Printing Office and the process of printing legislation.* The introduction of camera ready processes has significantly reduced the role of the Printing Office in the printing of legislation. Whereas previously they had to create the final version of legislation, which was proofread by the OPC, the OPC now provide them with a camera-ready version.
- *changes in how government agencies and the public access legislation.* Although public access to legislation was divided into a separate project, people became aware that consolidated legislation was potentially available and, when the project became delayed there was considerable frustration expressed. The implementation of publicly accessible legislation allows lawyers to reduce costs by eliminating the need to maintain manual paste-ups of legislation and enabling electronic text searching. This could impact on the roles of individuals who have been responsible

for such paste-ups. In some cases, the maintenance of such paste-ups was a large part of their role.

These changes, although significant in their own right to those involved, are minor compared with the potential changes which could emerge over time. These potential changes include changes to:

- *the methods by which Parliament and instructing agencies receive legislation for review.* If they receive draft legislation or Bills presented in a strikethrough/underline format, they will be able to more easily see the consequences of their decisions²⁰. Depending on the details of such changes, it could feasibly change the nature of debates in Parliament.
- *the consolidation of Tasmania's legislation* has emphasised the untidiness of some statutes and they may be reviewed. This was the first time most of Tasmania's 639 statutes have been consolidated since 1959. The consolidation activities picked up a number of errors, many of which were simply grammatical, but some which were significant (one Act was found to refer to a minister of faeces after amendments had been consolidated). This consolidation exercise may stimulate changes to the statute book.
- *improved access to legislation by lay people*, as specific legal expertise and considerable time is not required to consolidate legislation. Printed versions of Tasmanian consolidated legislation progressively became available via the Government's printing office. By mid-1997, there were 86 acts available in an "unauthorised" form, being released before the commencement of the new *Legislation Publication Act 1996*. By the time the EnAct system was ready to be implemented in late 1997, 571 acts were released in this way.

While initially, the plan had been to recoup the costs involved in the LSP by selling consolidated legislation, consolidated access was now to be provided free, as part of the government's responsibility to provide citizens their right of access to the law. These changes are reflective of the emergent nature of systems development.

- *greater efficiency by government agencies and legal organisations, who are no longer required to maintain their own pasted up versions of legislation.* Units such as the Audit Office and Police spent considerable effort in maintaining their own copies of legislation and this will now no longer be required. This may entail that the roles of legal clerks in such areas may change.

²⁰ See appendix 4 for an explanation of strikethrough/underline format.

As outlined in the LSP business case, these changes could improve the efficiency of parliament and government agencies and legal firms and lead to improved access to the law. However, while the LSP helps enable some of these changes, it will not induce them directly. It is difficult to provide firm details at this stage as changes are still unfolding, but it does give an insight into changes which could be negotiated via the LSP. It also indicates that change should not be viewed as a “one shot” process that is definable in a single project.

3.6.2 Nature of work processes (technical and process infrastructure)

During the early to mid-stages of the project, those involved foresaw that the project could be associated with major changes in the OPC's workprocesses. Examples of procedural changes identified by the project's Impact Analysis Report include:

- Amendment legislation will be performed by using the drafting tools within the legislation system to “mark up” the relevant principal legislation.
- All drafts will have to be modified into the correct SGML structure, regardless of the method used to draft it. That is, legislation can be created without structure, but that structure will have to be added to meet the requirements of the system.
- Drafters are able to electronically search consolidated legislation.
- Some drafting tasks, such as repealing or revoking legislation will only be able to be completed using the system.
- Indexes of Bill and Statutory Rule numbers, Acts, administrative summaries of all acts, cross references contained in Acts and so forth, previously maintained by OPC's records clerk, will probably not be necessary with the new system.

Many procedural issues had not been resolved by the time the system was implemented and this report merely highlighted them for consideration. An analysis of the future impact of the EnAct system remains an area for future research. Some procedural changes associated with the already implemented camera-ready processes (see Chapter 4.3) included no need to proofread printed documents or change documents into a different electronic format, tighter coordination and tracking of amendments between the OPC and Parliament and improved facilities for drafters to view the format of legislation. Other procedural changes were outlined as part of the Impact Analysis Report, part of which is included as Appendix 3. However, many of these are unresolved at the time of writing and are likely to remain so until the OPC have been using the system for some time. This reflects the emergent nature of the systems development process and the fact that the results of systems development projects do not simply unfold as soon as the technology is implemented.

One of the most significant changes associated with the introduction of the new system was standardisation, especially in the wording of amendment legislation. Standard wordings were required if the new system was to automatically produce amendment legislation from the consolidated legislation in which the drafters marked desired changes. Many discussions were held during the detailed design stage to determine what the appropriate standard wordings would be. Further examples of standardisation in relation to the LSP are provided in Table 3.6.1.

Date	Examples of standardisation associated with LSP
6/9/94	Standard interim formats for drafts discussed, and standard amendment wordings.
4/11/94	Agree on standard letters and forms, glossaries, macros, templates.
	LSP based on business rules.
14/11/94	Development of standard forms for OPC.
9/12/94	Process charts for camera ready processes- though deputy chief drafter says these do not define "thou shalt do this". OPC now only accept maps and charts in particular formats.
27/1/95	Project manager says drafting will become more standardised with the new system (though later recognise need to override system 23/8/95).
1/9/95	Standardisation of statutory rules structures.
22/11/95/ 20/12/95	Standard wordings for amendments discussed; CPC comments that "The name of all this game is we are trying to build some consistency into the way we do things".
5/9/97	Impact analysis report: "EnAct provides a series of steps which represent the logical work flow for the preparation and processing of legislation..." "The entering of the legislative text can be performed in a creative "free form" manner, or by adding each piece of legislation in its final format... Regardless of the method used to draft the legislation, it needs to be modified into the correct structure before it can be converted into SGML" (pp 24-5). "Legislation is loaded on the consolidated database by a commencement utility... As the integrity of the legislation database is based on the commencement dates, the process needs to be completed accurately" (p 31).

Table 3.6.1: Evidence of standardisation associated with the LSP.

At times standardisation was downplayed. For example, the deputy chief drafter explained that business rules defining OPC's current and possible future workprocesses in broad detail in the Functional Requirements Document did not define "thou must do this". When discussing standard wordings for legislation, the project manager commented the OPC were already using standards and so nothing was really changing. For example, they used the Macquarie Dictionary as a standard (22/12/94). The executive manager also commented that drafting was 80% creativity and the LSP was only aiming to improve the effectiveness of the other 20%.

Nevertheless, the EnAct system promoted standardisation. Wording for amendment legislation were particularly standardised and deeply embedded in the working of the new system, though these could be overridden. In other areas, the project promoted standardisation but did not enforce it. Through the analysis of OPC business rules, OPC standard practices were defined.

The link between standardisation and computing has been noted, with Boland (1995) commenting,

Information technology has a kind of magical power attributed to it for sharpening edges that were dull, making clear what was blurred and purifying what was contaminated (Boland 1995: p 321).

Thus, observations of the LSP confirm other studies. The introduction of computerised technology signals a commitment to systems thinking and systems engineering (Orlikowski 1988). This issue also reflects style of the quality management principles adopted by the LSP team, with Jenner (1993) commenting,

In order for our organisation to function effectively it is essential that we define the rules, roles responsibilities and requirements of our processes, as well as our expectations and work standards. Once we have done this, we can leave people to get on with their work. In other words, we have the right environment for delegation. If we go further, we can truly empower people by defining the desired results, providing the guidelines within which we want people to operate, identify the resources available for their use, describe the accountability and control mechanisms and agree the benefits to be gleaned by the organisation and the individual (p 12).

In other words, empowerment is said to derive from control through standardisation. Orlikowski (1991) observed that control need not just exist through social interactions, but can be embedded in the technological infrastructure through policies and so forth. In this way, IT is a medium as well as product of social interaction and expert authority can become embedded in standard practices which are governed by those with hierarchical authority.

3.6.3 Role changes

Burris (1993) suggests that computerisation is associated with greater distinctions between expert and non-expert sectors, and so reinforces existing technocratic tendencies. Yet there is little evidence that such bifurcation is occurring or will occur in the OPC as a result of the LSP. With the implementation of EnAct, they predicted that people in the office would become more multiskilled, and divisions between the two groups could be reduced. At this stage there is little evidence of this occurring, except in the role of the new systems administrator (ex-executive officer). This may be because, prior to the LSP, while there was a strict division between the drafters and the administrative staff, most of the administrative staff were considered to be experts in particular areas. The administrative staff generally had specialised skills regarding the structure and format of legislation. When discussing keyboarding, several drafters recognised the skills of the administrative assistants in this area, and regarded them as experts. The LSP and the EnAct system seems to be broadening their role at the same

time as removing this area of specialisation by providing keyboard training to the drafters. Thus observations of the LSP do not back up Burris' observations in this respect.

The most fundamental change resulting from the LSP will be written legislation in a form which allows easier access to both consolidated and unconsolidated versions of acts as acts and amendments are linked electronically. On the surface, this would suggest that the way the drafters create legislation could substantially change and that the roles of the administrative assistants would become redundant. Yet this is not necessarily so. It is possible for a drafter to not change the way they work greatly, so that the role of the administrative assistants expands to include the "marking up" of drafts into SGML formats (ie, including structural tags in the writing). Some of the systems developers have anticipated some drafters may not change their workpractices with the implementation of the new system. Several drafters commented that they will still rely on the administrative assistants to complete large quantities of typing (for eg. when drafts are being adapted from other state's legislation) or to mark up and convert the drafters' work into the required EnAct structures. Many of them now write using the EnAct system, at least to a degree, but at this stage it is too early to firmly say the system has changed the roles of the drafters otherwise. While the systems developers anticipated the drafters would now complete certain functions (with the click of a key) which administrative staff members use to (take considerable time to) do, several drafters expressed resentment that they were being paid to perform administrative work. Such possible role changes are still emerging and remains an area for future research.

Barley (1990) suggested organisational change cannot be seen to have occurred if there are no role changes. The roles of some of the support staff are changing or will probably change. The Executive Officer is taking on the role of system administrator. The roles of the administrative assistants and records clerks remain unclear at this stage, and will depend on how the drafters utilise the EnAct system. However, it is apparent that their skills will still be required. Far from being threatened with job loss, the administrative assistants are being required to both input existing previous legislation ready for the formal launch of the EnAct system and support the drafters as they create current legislation. They have been doing this while learning to use what is widely agreed to be a very complex and sophisticated system.

The relationship between the administrative assistants and individual drafters is particularly fluid in the months following the implementation of the system. In the short term, at least, they will continue to type the drafters' work, but this may change as drafters increasingly type their own work. The drafters may format their work according to SGML standards or may ask the administrative assistants to do so. After

the system was implemented, several drafters commented they found it difficult to use the system directly for some types of drafting and that it was an inefficient use of their time to enter any work they had done by hand. It will probably be some time until firm role changes become apparent in the OPC. Essentially, role changes are still being negotiated as OPC staff members learn how to use the system and it becomes embedded into organisational processes.

3.6.4 Changes in the formal authority structures

At various times during the project, those involved claimed the LSP would empower the staff of the OPC to have more control over their work, and would give the management of the OPC more control over the work of the staff. As paradoxical as this seems initially, both predictions seem to be occurring to a degree.

The systems developers believed that the introduction of the EnAct system could help break down the strict divisions between drafters and their support staff, as the support staff gained specialist skills and people in the office became more multiskilled. They believed that the support staff would probably be responsible for enforcing management's standards, but recognised that the support staff could not do so without management's backing.

The LSP has heightened concerns as [the non-drafting staff] feel they will require a higher level of support by management with their future roles and activities (transcripts 3/8/94).

With the implementation of the system, the executive officer took on the role of systems administrator, a position that was to give him greater authority, due to his expert knowledge about the system. Several other staff members gained authority by having knowledge about all or some aspects of the system and there were some comments from some of the drafters about this. In some ways, therefore, the LSP has helped break down the strict separation between the "specialist" drafters and their "support" staff.

The systems developers also predicted that with the implementation of general computerised tools, such as email, the management of the OPC would not be able to scrutinise all the interaction OPC staff members had with other people, as such communication was physically less open to observation. At the same time, the systems developers were able to answer OPC management's concern that the individual writing and work styles impacted on the quality and consistency of the OPC's output through EnAct's reliance on standard wordings and processes. With the increased standardisation embedded in the EnAct system, the management of the OPC could impose standards on the staff of the OPC and so broadly control their work practices and outputs through these standards. Of course, though, most staff had some input into the creation of these standards through review meetings and the

interpretation of these standards. That is, control occurred not only from the top of the office downwards, but over time, as people helped developed standards they would later apply themselves. This issue is pursued further in Chapter 4.3. The new system also provided reporting facilities for the chief drafter to obtain information about drafting files within the office and workflow tools to monitor the progress of each legislative drafting task (Impact Analysis Report v 1.0 5 September 1997). Thus, the LSP can be associated with changes in the OPC's authority structures.

The development of the system helped define what people's particular areas of expertise were. For example, during debates about structuring work processes it was clarified that, while the management of the OPC could impose standards on the content of their work, the process by which drafters created that content could be largely decided by the drafters themselves. That is, the drafters could decide how they drafted the legislation, as long as they met the standards set by management (Impact Analysis Report v 1.0 5/9/1997).

The OPC is an organisation attempting to resolve the commonly occurring conflict between hierarchical and expert authority and the beginning of the LSP happened to coincide with a time when there was a high level of conflict within the office due to these issues. These issues seemed to have been at least partly resolved by the end of my involvement in the project, but it is unclear if the LSP and the resulting EnAct system assisted with this. However, the discussions surrounding the LSP did help resolve some of this conflict as hierarchical authority was enforced in some cases and areas where individual expertise was relevant were defined. This illustrates that authority structures are negotiated on an ongoing basis and are subject to change.

3.6.5 Cultural changes

Perhaps the biggest change introduced with the LSP was change itself. The OPC was a relatively stable organisation whose work processes had not changed for a some time (transcripts 14/12/1994).

It was quite difficult to track specific cultural changes within the OPC. However, the close interaction between the OPC and the quite different CIPU aided an examination of existing norms. At times the CIPU extended their own cultural norms into the OPC. For example, the CIPU organised some social events and promoted general discussions to review decisions made and existing practices. They also heavily emphasised the importance of documentation and systematic processes, as discussed in section 3.6.2. At this stage, it is difficult to see if these imported norms will continue and if changes in the OPC's authority structures will result in cultural changes. Certainly, some OPC staff members felt the CIPU were promoting their own approach and they resented this, viewing it as "cultural imperialism". On the

other hand, others expressed the view the OPC were quite inward looking and, as the office's heavy workload precluded OPC management's ability to focus on the quality and efficiency of OPC procedures and practices, it was beneficial to have some external input.

3.6.6 Summary and analysis

The formal scope of the LSP, as outlined towards the end of the project, gives an insight into the changes that were planned, or negotiated to be part of the project. These, along with areas which were considered, but were defined as outside the scope due to time, resource or other reasons, are listed in Table 3.6.1. In general, most of the organisational changes resulting from the LSP are to directly impact primarily on the OPC, and possible impacts of the project to other organisational units, apart from the printing office, are limited. Many areas that are defined as outside the scope of the project, such as the uploading of subordinate legislation and the correction of integrity problems encountered while consolidating legislation, are not covered due to a lack of resources at this stage. It is hoped these areas will be covered in future.

Other activities, such as the provision of read-only access to the statute book, were defined as part of a related but separate project that is currently underway. Although originally defined as part of the LSP, and used for marketing the project, a lack of funds and the already great complexity of the LSP promoted the definition of a separate project, the Legislation Publication Project (LPP). It is important to note these documents were originally intended to be within the scope of the project, as was the provision of public access to the document database. These, and other changes in scope reflect the emergent nature of the process as circumstances unfold.

In summary, while some of Orlikowski's (1988) predictions also seem to apply to the changes induced in the OPC by the LSP, others do not. Table 3.6.2 summarises these observations. Some of the differences between these observations and those of Orlikowski's be attributed to the differing organisational contexts being examined, the ten years difference between when the two studies were conducted and the slightly different foci of the two studies. While Orlikowski's study focused on issues of power and control, these issues relevant, but not central to this study. This study backs up Orlikowski's compulsions that computerisation can be associated with moves towards technical rationality, increased formalisation, abstraction and reification of task processes and greater standardisation.

Orlikowski's observations focused primarily on issues of control and computerisation. While this is an integral theme revealed by this study, the focus is slightly different from Orlikowski's. While she focused primarily on control and the outcomes (content) of computerisation, here the focus is on control and the process of change.

Specifically, this dissertation asks, “To what degree can we control or influence others and future events?”

In many cases, the consequences of the LSP partially outlined above were not completely predicted, planned or controlled. Many of them emerged as events and circumstances unfolded. This is due to the ongoing influence of the context and the social processes described in Parts 2 and 3. By definition, the context includes elements which are outside the control of those involved in particular changes, and these elements may have an ongoing influence. The social processes surrounding systems development involve the active involvement of participants through the ongoing negotiation of meaning and interest and the creation and sustainment of coalitions of meaning and interest as well as the application of expertise and authority. These issues all influence both the changes produced as a result of the LSP and the process by which they can be achieved. The process by which changes can be planned and controlled is the focus of Part 4.

Objectives that are part of the LSP (inside scope)	Objectives that are not part of the Project (outside scope)
<ul style="list-style-type: none"> • A legislation drafting and consolidation subsystem for the OPC which includes the following facilities: <ul style="list-style-type: none"> • shared file access; • editing and version control; • audit trail for changes; • draft status tracking facility; • formatting to meet Tas. Legislation presentation standards; • an interface to legislation data base and text retrieval subsystem; • a facility to allow amendments to be made to Bills during their passage through the Parliament; • automatic consolidation upon enactment; and • security from unauthorised access. 	<ul style="list-style-type: none"> • Not about providing a legislation drafting facility accessible to people who are not employed within the Office of Parliamentary Counsel. • Not about changing the way agencies and the Cabinet Office provide instructions to the OPC. • Not about providing a general administration records tracking facility.
<ul style="list-style-type: none"> • A Legislation database subsystem with: <ul style="list-style-type: none"> • access to a facility to enable the conversion of existing legislation stored in hard copy and magnetic tape form to a format which enables the legislation to be stored and consolidated in the legislation database; • verification procedures to ensure accuracy and completeness of uploaded legislation; • a facility for recording errors encountered during the conversion process occurring as a result of pre-existing integrity problems with the source data; • storage capacity for all consolidated Tasmanian Acts and Statutory Rules; • a text management system with index, search, retrieval and version control to provide access to Acts and Rules; • an index of Acts and Rules held on the database; • a facility for retrieval of individual Acts or Rules; • a search and retrieval facility over all or part of the database; • history of changes to Acts and Rules from the implementation of the subsystem; and • recreation of Acts and Rules at a particular time from the implementation of the production system. 	<ul style="list-style-type: none"> • Not about providing a database containing legislation from other States, Territories or the Federal Government. • The uploading of subordinate legislation is not an objective of the first phase of the project. • The conversion of existing legislation does not include the correction of pre-existing integrity problems encountered in the Tasmanian Statute Book. • Historical versions of Statutory Rules will not be loaded onto the database. • Repealed legislation will not be loaded onto the database. • The exclusion from the database of Subordinate Legislation drafted by authorities other than the OPC (Local Government By-laws)? • Recording history from the enactment of amending legislation if the amendment occurred prior to the implementation of the production system.
<ul style="list-style-type: none"> • A communications network that provides: <ul style="list-style-type: none"> • access to legislation drafting subsystem and legislation database subsystem inside and outside normal working hours, and the OPC office environment. • an interface to printing facilities in PAT;. • access to a "mirror copy" of the legislation database for third party providers of legislation value added products 	<ul style="list-style-type: none"> • Not about providing other potential users of the legislation database with read only access to the database. • Not about the implementation of new systems to provide copies of consolidated Acts and Rules for sale (eg. bound hard copy, loose leaf, CD format) - this is a related project.

Table 3.6.1: The planned scope of the Legislation System Project (from LSP Business Case Version 0.A 21 November 1997)

The nature of production tasks	
A shift towards technical rationality	Yes- through greater standardisation, but not completely, as the expertise of individuals is acknowledged.
Increased technical and cultural control	More technical control through standardisation. No evidence of major cultural changes noted.
Deskilling of production tasks	Administrative assistants loose specialist as keyboard experts but remain important in overall workings of office.
Increase of formalisation, abstraction and reification of task content	Yes- through use of standard wordings and SGML structures.
Integration of the division of labour	None observed, but nature of relationship between drafters and administrative assistants still fluid.
The expertise associated with the production processes	
A synthesis of technical expert and managerial roles, achieved through joint project teams or technical managers	Already existing.
A conflict over functional and technical territorialism	Yes, to degree, but functional authority justified with reference to different technical or professional expertise between the CIPU and OPC. Within the OPC, conflict between these two sources of power reduced slightly as individual areas of authority are defined.
The production strategy underlying the production process	
A shift towards generalised problem-solving approaches	Possibly but not necessarily. The EnAct system does not prescribe a method for drafting, and the structures required by the system can be enforced towards the end of the drafting process.
Increased programmed customisation. That is, the production of services by standardised problem-solving logic.	Not observed.
Organisation of people around the production processes	
Increased debureaucratisation, with more client participation, multi-disciplinary project teams and loosely coupled semi-autonomous production units;	Potentially - not observed at time of system implementation.
Decentralisation supported with elements of centralisation, as control is centralised	Yes- through standardisation.

Table 3.6.2: Orlikowski's (1988) predictions on computerisation and control and the LSP

4. Process

– IS planning, development and implementation

Let us start with an observation from Part 3: when the Legislation Systems Project was being defined and planned, the actions of those involved broadly aligned with rationalistic, normative assumptions. However, when the design and implementation started to be considered, these models were no help. Implementation problems were just that according to the literature: "implementation problems". They were not a reflection on the plans made but on those responsible for the implementation and defined as issues of "change management". Here this stance is examined in its wider context of normative conceptions of developing systems and the process is viewed at a macro and institutional level.

A process model should convey the idea of movement over time, like the description of a journey. It is generally recognised that there are very few good models of the process of information systems development and implementation (Walsham 1993). Particularly, they rarely illustrate how the process of change is tightly interwoven with the content and context of that change.

Process models, such as Walsham's, focus on systems development as social interaction and usefully give us an insight into the process at a micro-level. However, they only partly suggest what the process might look at a macro-level, even though this is the level at which systems developers consider their future actions. The focus of Part 4 is on how the overall process of systems development is conceived. After illustrating fundamental problems with the dominant systems development lifecycle (SDLC) model, alternatives will be examined which, by reflecting chronological and hierarchical emergence, do not suffer from the same criticisms. This, in essence, focuses on the *dasien* level of temporality.

The final section focuses on the institutional level of change, or the *longue durée*. It will illustrate that the change involved with systems development projects can be viewed in terms of structuration theory and the ongoing negotiation between individuals' realms of action and the structural frames they perceive. Building on the previous sections, Part 4 will describe and justify a multi-level model of the process of systems development.

4.1 Change management and its relation to systems development

“Change management” is an often used phrase and concept in relation to system development projects. An examination of the use of the term in both literature and practice reveals confusion as to what is precisely meant by it. Three distinct, though inter-related, interpretations of the term can be identified and are examined. This chapter will also illustrate that a commonly used interpretation of the word embodies a problematic conception of the systems development process.

A well-respected consultant stood in front of a group of Tasmanian State Service project managers. “Change management is an important issue for many projects”, he commented. “And a problematic one. Frankly, I don’t know of any firm or good solutions to it” (transcripts 6/12/1997) The term, “change management” is often referred to in relation to systems development projects and is commonly used in practice. Three distinct interpretations of the term can be identified both in the literature and practice. This not only indicates a confusion of terminology, but reflects fundamentally quite different ways of approaching change in organisations.

Managing change effectively is crucial, given the rapid changes in the context of most organisations. Buchanan (1993) comments that many see the implementation of change as a critical issue while Berger (1994) suggests it is an emerging management discipline and profession. At this stage, the definition of change management is necessarily vague as most sources are nebulous and there is confusion as to precisely what the term means. Here the use of the term “change management” is investigated in both the available literature and in the LSP case study where the term has been used in practice. Three distinct, though inter-related, interpretations of the term can be identified and are examined.

4.1.1 The ubiquity of change

“Change is inevitable, except from vending machines”. According to the literature on change management it is happening all the time and we must manage it before it manages us (Berger 1994). Traditional management is geared towards managing in stable contexts. With today's world, however, there is the need to manage responses to rapid changes in the environment efficiently and effectively if organisations are to survive. The implementation of new information systems, most of which involve the use of sophisticated computerised technologies, is indicative of these trends.

Despite, or because of, the ubiquitous use of the word, there is a great deal of confusion as to what we mean by “change”. As a word, it can have two related, but

very distinct meanings. As a verb, it refers to the process of changing but as a noun, it refers to an outcome: the changes that have or are to be implemented. Much of the confusion surrounding the concept of change management can be linked back to these two aspects of the same word. Are we managing the process of change, or the outcomes (content) of it? When do we determine what the outcomes of the changes shall be and who makes the decision? How planned are changes and when are these plans made?

There is also confusion as to how we describe change. In particular, what do we mean by a high or low degree of change? Strauss and Corbin (1990) identified some of the possible characteristics of change, as listed in Table 4.1.1. "Radical" change could refer to fast, unplanned, random, forward, upward, wide and/or great change. It could also refer to the degree of change or its speed (Legge 1984). Thus, definitions of "radical change" are often vague and open to several interpretations and reveal confusion as to what is exactly meant.

properties	dimensional ranges	
<i>rate</i>	fast	slow
<i>occurrence</i>	planned	unplanned
<i>shape</i>	orderly	random
	progressive	random
<i>direction</i>	forward	backwards
	upward	downward
<i>scope</i>	wide	narrow
<i>degree of impact</i>	great	small
<i>ability to control</i>	high	low

Table 4.1.1: Characteristics of change (Strauss and Corbin 1990)

Several authors have suggested that there are two fundamentally different approaches to change in organisations (eg Dawson 1994, Burnes 1996). The first views change as a planned, rational and largely linear process, which reflects strong top-down tendencies (Burnes 1996; Buchanan, 1993) while the second emphasises individuals' goals and political issues, the untidiness of change and its often illogical nature and is conceptually aligned with emergent or evolutionary policy planning models. While the first is aligned with rationalistic assumptions, the latter suggests an alternative.

a) The planned, top-down approach to change

Top-down approaches generally emphasise the need to carefully plan, with change management tasks aiming to implement these already established plans. Hence, there is a real split between establishing the content of the change project and the process of implementing it. The top-down approach to change management emphasises management's prerogative and the need to overcome resistance. Such approaches generally use Lewin's three stage model of change as their basis (Dawson 1994).

Lewin suggested that successful planned change programs essentially include a stage where the status quo is "unfrozen" before change takes place and is "refrozen" afterwards. This model provides a general framework for understanding the process of organisational change, but the steps are very broad. Hence there have been many attempts to further articulate the phases to aid their practical applicability (Burnes 1996: p 183). Reviewing over thirty models of planned change, Bullock and Batten (1985 in *ibid*) produced an integrated four stage model:

- | | |
|--------------------------------|---|
| 1. exploration | becoming aware of the need for change, searching for outside assistance (consultant) |
| 2. planning | collecting information to establish "a correct diagnosis" of the problem, establish goals of the change and design actions to achieve these goals, obtaining support from key players |
| 3. action | establishing appropriate arrangements to manage the process, gaining support for actions taken, evaluating the implementation activities so refinements can be made. |
| 4. integration (consolidation) | reinforcement of behaviour through feedback and reward systems, gradual decreased reliance on external assistance, diffusing the change throughout the organisation, training to continually monitor changes and look for improvements. |

Such top-down models of change have a number of important characteristics:

- The content of the change program is established up front and change management activities aim to ensure their implementation. Note the division between planning and implementation (action) activities above.
- Management prerogative is emphasised, with communication activities emphasising a need to communicate *to* others rather than with them (eg Davidow 1994).
- These activities are justified through references to the external environment of the organisation and "resistance" is seen as an important obstacle to overcome. Opposition is seen as irrational (eg Berger 1994).
- These planned approaches emphasise the role of an external facilitator so that an "unbiased view" and non-aligned stance can be taken (eg McCalman 1992).

The top-down approach to change is common yet it is based on some very questionable assumptions. Firstly, the model is extremely rationalistic, reflecting assumptions that people can be omniscient, values and personal perspectives can be ignored and that all in an organisation have the same goals. These assumptions have been strongly rebutted in previous chapters. This model assumes that the changes planned *inter alia* are good and justified. The idea that some plans are dumb ideas or that they only benefit certain sub-groups is not entertained. The top-down approach to change management can be broadly described as activities designed to get *them* to do what *we* want. The content/ output of the changes is established and the question change management activities have to answer is "how do we get them to support these changes so they can be implemented?" That the implementability of proposed

changes could affect the content of those changes is generally not addressed. These issues are worth examining in more detail and are illustrated with examples below. It will be argued in the next chapter that the commonly used model of the systems development process reflects these assumptions.

The assumption is that change needs lubrication to be implemented and change management provides it. It is seen as ways of ensuring *they* change in a way *we* deem appropriate, through the effective use of power and authority through communication channels and resources. Change management effectively becomes "ways of overcoming resistance to change". Differences of opinion are not tolerated (eg Worthingham Brighton Press 1997; Berger 1994). The importance of communication is often emphasised, but what they often mean by this is that changes need to be communicated *to* those affected so that often only passive involvement is encouraged. For example:

Organizations need to communicate to employees what the change is, how it will occur, and why the change is necessary (Becker 1997).

At other times, the need for two-way "open" communication is emphasised, but this conflicts with a need to "reduce resistance" and "create a shared understanding of the need to move forward". In fact, two way communication is seen as a way of achieving this. Only some writers openly acknowledge that the acceptance of a need to change by the majority of people involved is a crucial and unavoidable stage in the change process. Most do not make this explicit link between the process of change and its content, and assume that if a proposed change is abandoned because it does not have the support of others, then it is unsuccessful. Two-way communication is allowed, but only within certain boundaries with the actual need for change. Its general direction remains unnegotiable and the purpose of two-way communication is to develop detailed plans and obtain feedback on how the changes are operating. For example:

Oracle Change Management Services helps you develop and implement communication and feedback programs to achieve a unified vision and facilitate open communication among all levels of the organization. These programs reduce resistance to change, uniformly communicate benefits and objectives of the implementation, and create a shared understanding of the need to move forward (Oracle 1996).

That is, the purpose of two way communication is not to establish what the broad outputs of the change to is to be, but to facilitate the process of implementing previously established goals.

The importance of top management commitment is often emphasised, as is the need for a clear and unified vision (Davidow 1994; Warner Burke 1994). For example:

Rarely, if ever, does significant societal or organisational change occur randomly. Leaders step forward or are selected to do something about society's needs - for example Lenin, Lincoln, Gandhi, DeGaulle or Churchill - or to respond to an organisation's needs... (Warner Burke 1994: p 284).

In summary, change is seen to flow from the top of an organisation downwards. It assumes that "we" have the right to change "their" behaviour, with management prerogatives being greatly emphasised. There is a strict division between planning and implementation and communication refers to the provision of information and the exercise of authority rather than meaningful two-way dialogue with those affected by the changes. Aligned with the rationalistic assumptions discussed above, this approach suffers from the same criticisms. It is important to note this approach is common in the Tasmanian State Service. This is not surprising given the nature of the public service, and the requirement that they follow the directives of the politically elected government and often have limited input into decision making processes.

b) The emergent approach to change

Increasingly this planned, top-down approach to change has been questioned and alternative models suggested. These alternative approaches tend to be far more fragmented, with Burnes (1996) commenting that their main uniting characteristic is that they are critical of the top-down approaches described in the previous subsection. Several general characteristics can be identified, however:

- They stress the unpredictable nature of change. Change unfolds through the interaction of a large number of variables and such approaches tend to emphasise the importance of identifying environmental trends, threats and opportunities quickly then responding to them;
- While the pressure for change comes from external issues, such approaches recognise that how change unfolds depends on people within the organisation;
- The management of change is not a specialist activity driven or facilitated by an expert, but an integral part of all managers' roles;
- Such approaches tend to be more analytical than prescriptive, and claim that there can be few simple prescriptions. Specifically, change cannot be viewed as a simple series of activities and events from one state to another, but a continuous process. Managing change therefore depends more on understanding the issues involved and a range of available options for dealing with them, rather than the creation of detailed plans (Burnes 1996);
- Change tends to involve a large number of small-scale incremental changes which over time can add up to quite substantial changes;
- As the pace and complexity of environmental changes are too much for a small group of senior managers to comprehend responsibility for change is greatly devolved. Managers become facilitators of change, rather than drivers. Managers

are therefore not responsible so much for the actual changes, but providing others with the necessary skills and resources for adequately dealing with change. Hence change becomes more of a bottom-up process, and communication *with* all involved with the change area is emphasised.

Perhaps the most well-known author promoting an emergent approach to change is Pettigrew (1985; 1993). His exploration into change at ICI over 15 years gives a detailed description of the emergent process of change at that organisation. As well as the content (outcomes) and process of change, Pettigrew suggested that there was a third aspect: change does not occur in a vacuum but most writing on change assumes it does. Thus the context of change is an important ongoing consideration, these three elements of change are tightly interwoven and because of this, change must be emergent. One cannot establish the content of change and then begin on the process of implementing it, because the context keeps changing. As Pettigrew commented:

After all, if the present is capable of being seen in terms of crisis, complications, conflicts, contradictions and ambiguities why must the pathways from past to present be analysed in such a unitary, linear fashion? Except if in Crozier and Friedman's terms (1980:245) "such theories are ultimately no more than rationalizations useful for giving clear consciences to those who thus commit themselves to blindness" (1985: p 33).

This stance has obviously influenced this research project, and stands in contrast to the common top-down approach which influences systems development literature practice.

Criticisms of this emergent approach include:

- resulting models tend to be complex and difficult to apply normatively (Dawson 1994). However, while it can be difficult to apply them directly, these models can usefully inform practical approaches and models which are easier to use;
- they assume that the environment is unpredictable – this is not always the case (Burnes 1996). This criticism misses a major point that Pettigrew and others have made. The fact is that contextual elements tend to only gradually emerge as people gain an understanding of the situation, and that is largely why the environment is unpredictable. The fact that it is also changing is only one aspect of the issue;
- they tend to emphasise the need to create an appropriate culture of change and the importance of building a "learning organisation, ignoring the fact that such organisations are difficult or even impossible to build due to power and status issues involved (Burnes 1996). Both Pettigrew (1985) and Dawson (1994) acknowledge the importance of authority and the influence of external issues in change;
- The greatest challenge of the emergent approach to change is to managers themselves, who are asked to change their approaches. They may not accept such changes, especially if it does not align with their views on what works (Burnes

1994). This is a valid criticism, as an emergent approach can represent a significant shift in thinking for managers and systems developers alike.

Thus, there are some valid criticisms of this emergent approach to change management, though they tend to be less fundamental than those directed at the top-down approach.

c) Two distinct approaches to change

After criticising both approaches to change and examining several case studies, Burnes (1996) suggests an alternative model to the top-down and emergent approach. This model accepts that while change can be complex, ambiguous and open-ended, it can also be relatively straightforward. Essentially, Burnes' model claims to combine the best elements of both the top-down and emergent approaches to change. The basis of Burnes model is that managers have a considerable degree of choice when they make decisions while other models assume they are only able to react to and align with their organisation's environment. Planned change views change management as a way of producing a pre-specified outcome while the emergent approach views it as a continual process of realignment with the organisation's environment.

Yet Burnes does not seem to have grasped that the top-down planned and emergent approaches to managing change are fundamentally quite different. The planned approach assumes that significant changes can be planned within the space of a single project: changes are planned, then implemented, with change management activities focusing on the implementation of these plans. The emergent approach to change disputes this. Pettigrew (1995) mentioned the need to not focus on projects as the unit of change, because change did not operate at this level. Change occurs over the span of a string or series of projects with projects implementing concrete goals established through emergent change management procedures. Burnes does not seem to have understood this when he views planned (top-down) and emergent approaches to change as compatible. They are different because their foci are different. This is illustrated graphically in Figure 4.1.1.

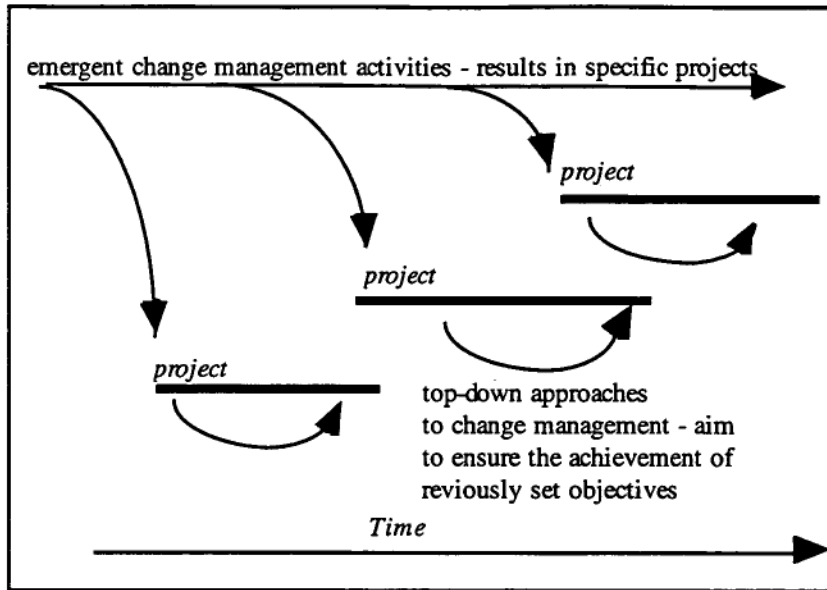


Figure 4.1.1: Top-down and emergent approaches to change - their differing approaches

Top-down planned approaches assume that change management is part of a project, and focuses on the implementation of previously established plans. Emergent approaches view change management as a broader process of identifying what changes are appropriate and how they can be implemented through projects. Through carefully and continually coordinating the appropriate people, emergent approaches to change management leads to the establishment of projects with firm goals. Thus emergent change management practices are essentially an integral part of planning activities and broader than the scope of single projects. Projects initiated as part of such a process are likely to be much smaller in scope and ambition because there is a strong recognition that the context has a huge and not entirely predictable impact on their success. Hence, how we view change management has significant implications for systems development.

4.1.2 Change management - one phrase with three interpretations

The identification of these two fundamental approaches to change in organisations suggests two quite different and distinct interpretations of the phrase, *change management*.

1. In very broad terms, the phrase could simply refer to the idea that change in organisations needs to be proactively managed. Project management and system development methodologies are used as tools or techniques *within* the area of activity referred to as change management. This interpretation aligns with the emergent approach to change identified above, in which projects are not viewed as the unit of change.

2. In more narrow terms, the phrase can refer within the scope of specific projects. Here the focus is on adequately ensuring the planned outcomes are achieved through being effectively embedded in their organisational context. This interpretation assumes a degree of separation between planning the content of the changes and the process of implementing it.

There is also a third widely-used interpretation of the term *change management*, which is distinct from these two definitions.

3. Changes to the planned outputs of a project are quite often required within the scope of the project. These changes and the responses to them must be effectively managed if the project is to remain on track and the proposed outcomes met. These activities are also sometimes referred to as configuration management or change control activities.

All three uses of the term *change management* can be identified in the literature on systems development projects.

The second interpretation of the word is the most common. There are many examples of writers assuming the content of change has largely been established, with an emphasis on vision, firm goals and a "road map". Overcoming resistance is seen as important part of change management activities, with Benjamin and Levinson (1993) commenting that "Change managers must work to unfreeze resistance. Only then can people consider new ideas" (p 29). The importance of top management commitment is emphasised through the role of a champion and management actions firmly at the centre of developments through the use of the MIT90's framework (Scott Morton 1991). Again, Smyrk (1997) suggests that a Change Management Plan should include communication to stakeholders, an assessment of risks regarding human factors, and intervention strategies. He comments that "sometimes you have to pretty hard" in carrying out these interventions. Educational texts report:

The ability to manage change is critical to the success of systems development... Unfortunately, not everyone easily adapts to change. Managing change requires the ability to recognize existing or potential problems (particularly the concerns of users) and to deal with them before they become a serious threat to the success of the new or modified system (Stair 1996: p 420).

Change management is therefore a very important skill for systems analysts, who are also *change agents*. You must know how to get people to make a smooth transition from one information system to another, giving up their old ways of doing things and accepting new ways (Hoffer, George et al. 1996: p 94).

[Participative, as opposed to coercive change management is] ... when the systems analyst educates the individuals affected by the new system. It is hoped that this introduction of new knowledge will cultivate the development of an appropriate attitude toward the new system..." (Burch and Grudnisky 1986) [Coercive change is initiated from a position of power aided by rewards, punishment and sanctions].

Note that people are expected to adapt to previously established plans. The idea that people's actions and attitudes have to be aligned with the proposed technological

changes is especially obvious in advertisements for consulting services on the internet. Oracle Corporation, for example, advertises what they term a "systematic, informed approach to change management",

- one which helps you manage the smooth, effective integration of Oracle technology into your business environment and ensures a successful transition to a new way of doing business... (Oracle 1996).

The concept of change management is commonly used in the literature on business process engineering as a way of addressing the common criticism that BPR is much easier to plan than to implement. There it can become broadly equitable with a marketing campaign. Hammer, for example wrote,

...change management lies at the heart of all reengineering programs. Few organizations possess the combination of skills, attitudes and experience needed for navigating a complex change through a maze of constituencies. Not many in-house reengineers have designed and executed a full-fledged communications program. Broad, vision-based marketing campaigns are rare, except in the consumer products industry - and even there, selling soap powder is different from selling organizational change (Hammer and Stanton 1995).

The argument runs that successful BPR also requires good "change management" processes if it is to be implemented effectively. The idea is that change management is an implementation issue, not a planning one. The same logic is being used in the systems development literature and by consultants in this area.

The fact that the use of the term change management in the systems development literature reflects top-down assumptions is not surprising given the top-down nature of the systems development literature (Korac-Boisvert and Kouzmin 1995). An exception is Mumford (1995), when she introduces the ETHICS methodology. Mumford's view of change management is more closely aligned with an emergent view of change management. For her, change management is not an activity within the process of systems development, but an effective systems development process is one way of managing change.

This first interpretation of the phrase is also reflected in the texts who describe project management as a method for managing change (eg McLeod and Smith 1996). Turner (1993), for example, talks of "managing change through projects". Projects are management tools for implementing changes. The Macquarie Dictionary defines a project as "something projected for execution, a plan, scheme, purpose; a proposal". Managing the realisation of such a plan is essentially a method of managing a particular change that is specified and given a specific scope.

The third interpretation of change management can also be identified. For example, Jordan and Machesky (1990) discuss change management as follows:

Change management: System changes are inevitable. The post-implementation evaluation may identify change requests that need to be acted upon immediately. If a system revision is required to meet the original project specifications, the work may be viewed as an extension of the original systems development (p 595).

In fact, two different interpretations of the term can be discerned, even within the one text. (eg. Hoffer, George et al 1996) While quite distinct from the other two interpretations, this third use of the phrase also raises pertinent issues. It highlights the fact that change, even within the scope of a single project, often cannot be planned or defined adequately up-front, and there need to be mechanisms for coping with emergent or unexpected issues.

4.1.3 Change management and the LSP

As has been indicated by the general literature, project management is seen as a way of managing change in the State Service generally. The increased focus on project management signals a shift from focusing on stable organisational processes. Thus, the first interpretation of the phrase *change management* was evident in relation to the project.

The systems developers were also aware that change management could also refer to the management of unforeseen or emergent issues, but generally termed these activities change control procedures to avoid confusion. However, they did still use the term to refer to changes in functional requirements and their management. For example:

A possible contributing factor to the 'change management' confusion has been our failure to clearly define the beginning and end of the prototyping phase. At this stage of the project, we consider the prototyping phase to be at an end and that any new functionality should be introduced as a result of a 'change request' (letter from project leader to technical systems contractor 6/5/96).

The use of the quotation marks around the term "change management" perhaps indicates an awareness that the term can have differing meanings.

Most of the time, the use of the term "change management" was used in relation to implementation. Change management activities revolved around the need to ensure the resulting EnAct system would be effectively utilised by the OPC and that the proposed outcomes of the project would be realised. In practice, many of the activities covered by the term change management were essential elements of any project, such as communication to and with those affected by the changes, training, adequate resourcing and other support. The approach also reflected the general assumption that the outcomes, or content, of the proposed changes had basically been established and that now change management activities were required to implement these changes.

In the early to mid stages of the project, those actively involved in the LSP anticipated great organisational changes to be associated with the project. During the planning stages of the project, the CIPU members became aware that the non-technical changes were a significant issue in the project, some of them commenting, for example:

- the implementation of the proposed system will involve a fundamental change to the work processes within the OPC. This needs to be addressed and the associated risks need to be identified, and the OPC must accept and plan for a significant change in their current work practices...
- The OPC aren't taking ownership of the problems that the project is addressing, therefore they won't accept ownership of the solutions, which introduces a high risk to the acceptance of the overall system (LSP status report 30/5/94).

I have trouble establishing a feeling of team involvement in LSP related activities. I'm concerned that all staff may not be sufficiently committed to the level of change which is necessary to the LSP. I have trouble establishing an open atmosphere in which I can successfully interact with OPC staff. I feel that internal Office issues that are outside the scope of the LSP are negatively impacting on the project (LSP project documentation 12/8/94).

The members of the CIPU thus saw the OPC as not open to change, with the project leader commenting publicly, and politely, that,

To ensure the successful integration of a leading edge technology into an area with rich and important traditions like the OPC, the CIPU adopted a rigorous development methodology (Technical System Developers' internal newsletter, Summer 1995).

Almost all the consultants involved in the project emphasised the need to consider how the system would impact on people's jobs and the implementation of these changes. These concerns provided the impetus for focusing on what was to be termed "change management".

It is difficult to track the exact origin of change management activities in the LSP. The identification of change management as a theme in the project cannot be wholly attributed to my involvement, as the systems developers referred to the concept prior to my involvement²¹. The systems developers were aware that organisational change issues involved in the LSP had to be considered if the project was to be considered a success. They sought information on change management to meet the perceived risks. The project director later commented that when others first talked of "change management" activities, he thought such issues were outside their area of concern as they were technical systems developers (transcripts 3/2/95) but generally he accepted that there was a close relationship between the success of the project and non-technical factors affecting the technical systems development project. A consideration of these issues generally came to be referred to as "change management".

²¹ for example: "We regard the system as having interim steps, with staged implementation into the OPC for change management issues" (4/5/95).

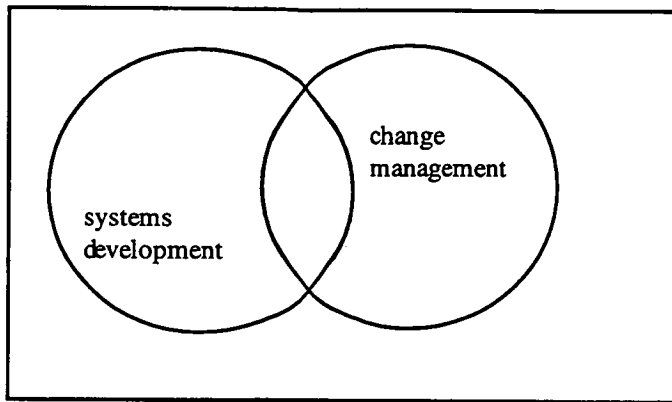


Figure 4.1.2: The relationship between systems development and change management as conceived by the second LSP project manager (transcripts 3/2/95)

However, initially there seemed to be some confusion as to what the concept entailed. The project manager described it in terms of a venn diagram, with information systems and change management in two interlocking circles, as is illustrated in Figure 4.1.2. This suggested she saw it as a separable activity from systems development. The project director referred to it as the management of non-technical aspects of the organisation including issues such as "management practices", "leadership", "culture", "mission and strategy", "task and skill requirements", "work unit climate" and so forth, as suggested by consultants at a training course. The systems developers put significant effort into finding consultants to help in this area, one of them commenting:

We realised that with our background and experience, we didn't have the skills to consider these change management issues and we've had to contract a lot of it out (transcripts 10/3/97).

In essence, they knew they had to do something, but could not determine precisely what. There was confusion about what the term precisely entailed throughout the project.

Later the LSP team defined change management, "all those things that need to be in place for the system to be successful", including training, documentation, the resolution of occupational health and safety issues and workflow/process redesign". They tried to proactively deal with all issues they defined as change management, but recognised that "a lot of change issues won't emerge until after implementation" (interview with project leader 10/3/1997). Specifically, they tried to tackle process change issues as early in the project as possible, but seemed to find this difficult as they did not have in depth knowledge of the OPC's work processes or details of how the EnAct system would be utilised by members of the OPC. In essence, they seemed to be trying to define implementation activities as a separable set of activities, so reinforcing a split between planning and implementation.

Concerns about the organisational change issues remained throughout the course of the project, with a later version of the Business Case commenting,

Significant change will impact on the culture of client areas, introducing a high risk to the level of support given to the project and the overall acceptance of the system (Business Case Document , v 4, 30/8/1996).

The document identified three strategies for dealing with these issues:

- involvement by OPC members in development, testing and other tasks;
- transitional steps (incremental implementation) with the implementation of camera ready processes being the main example; and
- the creation of a change management strategy within the OPC.

This strategy had replaced another quite different one:

A cultural analysis will be performed within the OPC to determine the impact of significant changes to the production and maintenance of legislation will have on staff [replaced 23/3/95].

This change is significant as it shifted responsibility for change management activities from the systems developers to those who would be affected by the changes. It resolved the systems developers' confusion about change management by abrogating responsibility for implementation and splitting planning, development and implementation activities.

However, during the final stages of testing the system, the CIPU were increasingly concerned the OPC still did not appreciate the changes that would be associated with the implementation of the new system and would be unprepared for it. In order to raise awareness of the issues involved and articulate them precisely, they developed or facilitated the development of a number of reports and plans focusing on the implementation of the EnAct system in the OPC. These reports included an Impact Analysis Report and a Change Management Plan. They still recognised that, even though they could not be responsible for the effective implementation of the system, the utilisation of the system would have a huge impact on the perceived success of the project.

They facilitated the appointment of another change management consultant who was to be responsible to OPC management to suggest tactics for approaching such issues. The scope of the consultancy included documenting the current processes in the OPC, documenting the processes *required* within the OPC to realise the benefits associated with the LSP, preparing a change management plan "for the transformation of processes within the OPC" and assisting the CPC with the implementation of the change management plan (transcripts 23/7/97).

Tactics recommended by the consultant included making and aiming for firm goals in terms of time and usage (for example, all staff to illustrate minimum proficiency in

the system within three months), in-house implementation support, extra staff during implementation, ongoing staff meetings and newsletters, user manuals, learner based training and “the development of a consistent rationale for change” by management. Overall, the consultant’s report includes a great deal of common sense advice. While very little of it would have been novel to those involved in the project, the fact that the issues had been articulated by an external party with perceived expertise in this area is significant. This provided legitimacy to decisions made and actions taken.

Hence, for most of the time during the LSP, “change management” referred to the second meaning of the word. It reflected many of the same rationalistic assumptions, but it did provide a useful category in which to place all the social and process issues the CIPU were having problems with and acted as a kind of “too hard” basket.

The formal definition of the phrase used by the LSP team also reflects the confusion surrounding it:

Change management: A set of techniques that aid in evolution, composition and policy management of the design and implementation of an object or system (Impact Analysis Report v 1.0 5 Sept 1997).

As the LSP project manager commented,

It is very difficult to discuss issues as complex as the topic of this meeting. Complex issues tend to be labelled with “*buzzword*” (eg Change management, Business Transformation, Process Reengineering etc.) yet these words have a different meaning for each of us (18/6/1997).

This confusion as to what change management meant, both conceptually and in practice reflects the general literature on the subject. In practice, it came to be translated into “anything which obviously has to be addressed but which is not covered by our existing tools and methods”. As such, it has helped to fill in the holes in existing system development methodologies and indicated problems with the underlying model of the process.

4.1.4 Reflections: Change management and systems development projects – confusion in a profusion of interpretations

The growing emphasis on change management reflects two trends in systems development. Firstly, it reflects the growing recognition that technological changes need to be adequately embedded in their organisational context. As systems developers and others realise that existing available concepts do not cover these issues, the concept of change management has been promoted. Secondly, the increased emphasis on change management reflects a move away from the strong emphasis on planning. We have become good at making plans concerning systems development, but increasingly recognise we are not so good at implementing them.

At the heart of confusion surrounding change management lies the relationship between planning and implementation. Are they necessarily sequential activities?

Does planning necessarily purely precede implementation? If a planned change is not implemented successfully, is it an implementation problem, or a planning one? Is change management concerned with the implementation of previously established plans or does it also involve the creation of those plans? Depending on one's interpretation, it could be either. Analysing the confusion surrounding the term "change management" identifies some inconsistencies and problems in how the process of systems development is generally conceived. This is the focus of the following two chapters.

Three distinct interpretations of change management can be identified in both literature and practice. One views change management as activities enabling the implementation of previously defined outcomes. An alternative emergent approach to change management includes the definition of change outcomes as part of the process. The third focuses on managing changes to a specific project. In other words, it focuses on managing changes to the *content* of a project throughout the process of development and implementation.

Stated in such terms, the second definition refers to managing changes being enforced in the *context* while the first definition focuses on the *process* of change. As Pettigrew (1985) noted, change in organisations involves three essential elements: content, context and process. Thus, depending on the interpretation used, change management could refer to the management of the context, content or process of change. Pettigrew (1985) suggests that the effective change involves all three elements. While the activities associated with each interpretation of the term are significantly different, the use of the same term to describe distinct activities is perhaps not surprising, given the close relationship between these three fundamental elements of change.

The focus on these interpretations of change management indicate problems with how we view the process of change. While the content of changes associated with information systems development and implementation have become increasingly sophisticated, both in terms of the technology employed and their use within an organisational context, the processes used to manage the changes have lagged behind and risk ossification. Reflecting the rationalistic assumptions underlying much of it (Korac-Boisvert and Kouzmin 1995), the systems development literature generally adopts the planned, top-down approach to managing the process of change. Within IS projects, change management focuses on the implementation of plans defined early in the process. When used in relation to the systems development lifecycle, it acts as a "string and sticky tape" solution to some much more deep-seated process problems. There has been a lack of emphasis on the management of the process of change, and this is the focus of Part 4.

In summary:

- “change management” is a commonly used concept in relation to systems development projects;
- an analysis of the literature and its use in relation to the LSP suggests at least three different interpretations of the word are being used; and
- confusion surrounding the term indicates problems in how we view the process of development, especially regarding the relationship between planning and implementation.

The following chapters focus on conceptions of the process.

4.2 The systems development lifecycle model of the process

The commonly used systems development lifecycle (SDLC) model conceptualises the process of systems development at a macro level as a waterfall where sequential steps are identified and, while iteration is permitted, the process is largely linear. The model has been widely adopted by the systems development field as a basis for both systems analysis and design and project management. Here, the model is examined and its use in the LSP is illustrated while three broad criticisms of the model are identified. These criticisms form a basis for identifying alternatives in the following chapter.

At a macro level, change is often described in a linear fashion where the process is illustrated by a series of identifiable steps. With systems development, such linear change models have become known as the systems development lifecycle (SDLC). This chapter will illustrate this model inadequately incorporates contextual issues and does not acknowledge the political micro processes of change nor the emergent nature of the process. Observations and analysis of the LSP also suggest that, even when it is used in practice, it does not well reflect the actions involved in the systems development process. That is, it does not adequately reflect the socially negotiated nature of the process at a micro level, as explored in Part 3.

4.2.1 The systems development lifecycle

The systems development lifecycle is a model of the process by which a computerised information system is planned, developed and implemented in an organisation at a macro level. The process is generally viewed as broadly linear, in the form of a series of tasks linked like pools in a river or in the motion of a wheel, as is illustrated in Figure 4.2.1. Iteration between these stages is recognised, but the general idea is that there is a movement from one task, or phase to another. As Friedman (1989) describes it, the life cycle is "an ordered set of activities which combine to make up the conception, development use and eventual replacement of new computer-based systems" (p 175). Derived from operations management, systems and management science literature, these models show a linear sequence of events to be followed, the main advantage being that the process can be standardised and well-defined (Eliason 1990: p 175).

Here the SDLC model refers both to the conventional waterfall model and the structured lifecycle model. Although they are sometimes separated, the differences generally only relate to the tools employed and the same criticisms apply (Krogstie 1995). Structured life cycle approaches employ graphical documentation techniques

to aid the specification of requirements in a top-down manner, including dataflow diagrams, entity relationship diagrams or other such semantic data modelling techniques.

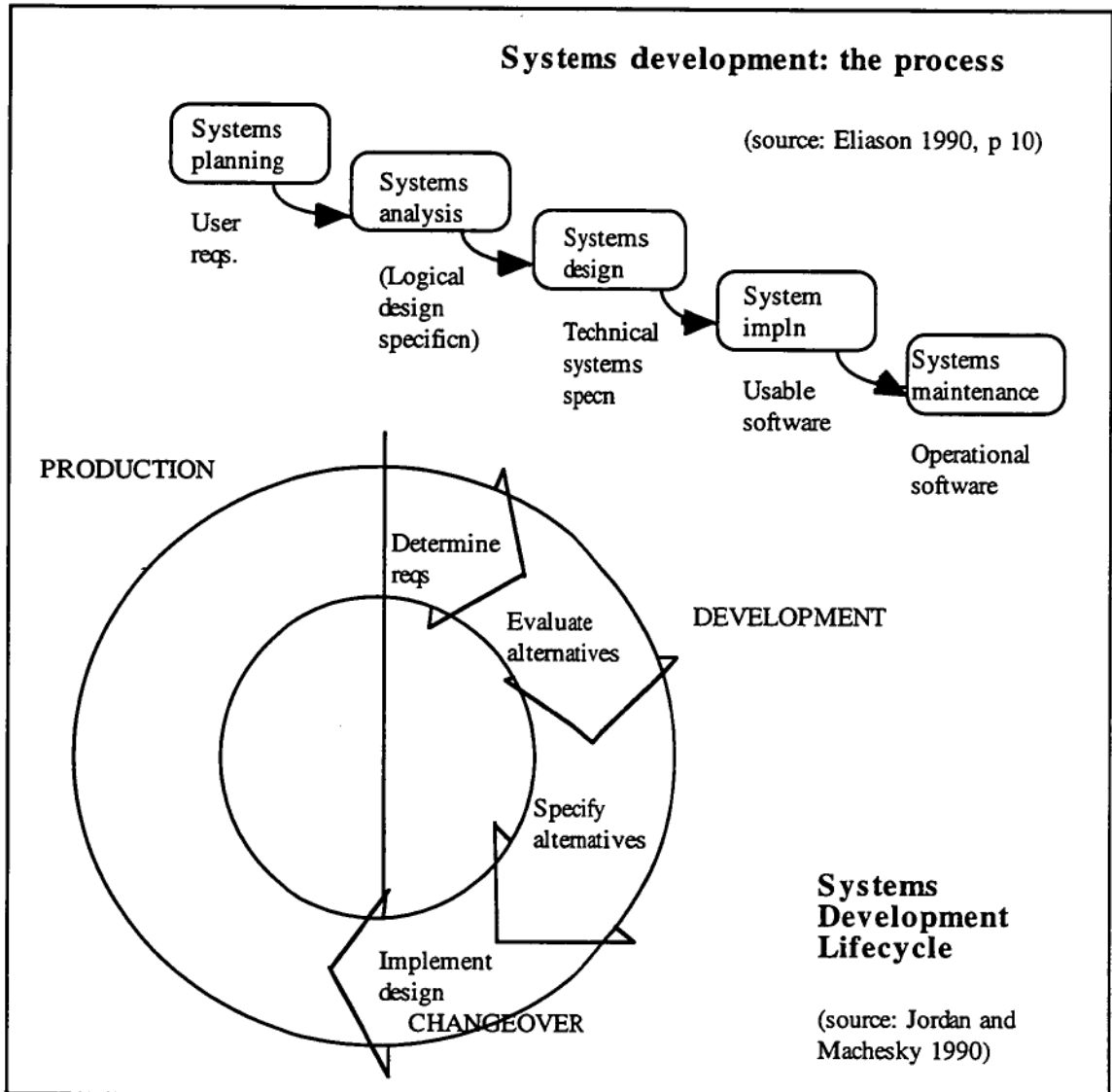


Figure 4.2.1: Two conceptions of the systems development lifecycle

As with general linear models of change, there is little consensus on the actual number of stages in the cycle²² or the actual tasks involved. However, they all have the same general linear structure. Friedman (1989) argued that changes in the SDLC reflected different concerns in the systems development process over time:

²² Eliason (1990), for example, uses the five stages of systems planning, systems analysis, systems design, systems implementation and systems maintenance, with the development process only covering the middle three phases. Eason (1988) provides another version of the lifecycle where the phases include project selection, feasibility study, systems analysis, requirements specification, systems design, construction, trials and implementation. Ahituv and Neumann (1986 in Buchanan and Boddy) described a ten stage model of the systems development life cycle covering the areas of: preliminary analysis; feasibility study; information analysis; systems design; programming; procedural development; conversion; operation and maintenance; post-audit; and termination.

We may think of the articulation of phases of a computer life cycle as a language for conveying what procedures are required for computerisation. Societies develop their languages in directions that reflect their priorities. We may chart changing priorities within computer system development by noting the changing extent to which different parts of the system lifecycle are articulated (p 249).

He described how an increased emphasis on software development issues produced a more detailed design section of the systems development process while the growing focus on user requirements has led to an elaboration of the requirements determination and implementation phases of the process. Although there are innumerable versions of the model, they tend to have the same basic structure (Miles 1985 in Lewis 1994).

Although initially strictly linear in nature, the systems development lifecycle has come to incorporate iterations, in recognition that it may be impossible to complete one task completely before going on to the next. As Friedman commented,

... strictly linear or waterfall life cycles have rarely been followed. Changes in general specifications often occurred once it was discovered, as a result of more detailed work, that the original ideas were unworkable or too awkward or expensive (the loopy linear pattern). Systems development is a creative process. To expect it to follow this waterfall procedure precisely is like expecting a novel writer to know precisely the plot and characterisation of his or her novel before it is written and not to change ideas in the process of writing (Friedman 1989: p 295).

However, the basic structure of the model has not changed and alternative approaches have not threatened it. Many methodologies today employ a structured analysis and design approach, where this cycle model is improved by tools which make the process more like an engineering activity, but essentially the process is still the same (Hoffer, George et al. 1996). Again, potential rivals, such as prototyping are now absorbed into the framework (Lewis 1994).

The systems development lifecycle is almost generally accepted as the norm for considering the systems development process. Most systems development texts employ it in some form or another, with 21 of the 25 texts on systems analysis, design and development surveyed using it as a framework (see Appendix 1). The model is used by Avison and Fitzgerald (1995) as a framework by which to judge methodologies, who state that the scope of methodologies "...is an indication of the stages of the lifecycle of systems development which the methodology covers" (p 455). The ISO 9001 standard for developing and supplying software systems views the process as a broadly linear one, with the process including the definition of functional requirements, programming, inspection and testing, and delivery and installation. Such international standards are used as a benchmark for assessing commercial methodologies and the process of systems development as it unfolds in organisations and Krogstie (1995) notes that most "standard" methodologies, such as SSADM are based on basic variations of the waterfall model. A methodology is "...a comprehensive procedural framework directed towards accomplishing a particular

change in an object system" (Welke 1983 in Hirschheim and Klein 1992: p 237) and it embodies concepts of process.

The idea that all information system development processes share a common natural structure is very attractive. Lewis (1994) comments that during the 1970s the idea that all projects proceed through certain stages in sequence became an ultimately unprovable but powerful and commonly recognised assumption. He argues that most research since then has focused on improving the process of systems development within this framework, commenting that the systems development lifecycle has "become so ingratiated in IS thinking that it is only rarely questioned" (p 68). Thus the SDLC model is often applied without question.

Echoing criticisms of the objectivist assumptions inherent in requirements analysis (section 3.1), Lewis comments,

The lifecycle idea is though more commonly introduced as a description of reality, and *as a fact* about systems development that the aspiring systems analyst and designer must know, much in the same way as a chemist should know that hydrogen and oxygen combine to make water or an aviation engineer should know that gravity pulls things earthward (Lewis 1994: p 68).

Smyrk (1997) described the SDLC as "the general anatomy of a project", where the whole project is divided into "discrete serial bundles of effort" and emphasised that nothing in phase B can start until phase A is completed. It is important to note that Smyrk provided training and advice to the CIPU and acted as a consultant during the later phases of the project, so these opinions influenced the LSP directly.

The lifecycle model does have strengths. It allows the overall task to be planned and subdivided so that the process is more controllable and, as it is predetermined, everyone understands the sequence of events so communication is improved (Eason 1988; Nandakumar 1993; Avison and Fitzgerald 1995). Avison and Fitzgerald (1995) argue there is intrinsically nothing wrong with the lifecycle model and much depends on the way it is used, commenting it needs to be sufficiently resourced and any deviations noted and controlled early. They argue it should not be seen as a rigid process, but a flexible and iterative one, though others emphasise the linearity of the process (eg Shelly et al 1995; Smyrk 1997).

4.2.2 Criticisms of the systems development lifecycle approach

Despite its wide use and the proposed advantages, the lifecycle model is recognised to have some serious and fundamental problems. Cited concerns include: an inability to cope with unexpected circumstances, user involvement and unanticipated results; a dependence on stable requirements and little recognition of the often emergent nature of change; an inability to incorporate pluralism or political rationality and very little consideration of micro-social processes; an expectation that users will have enough

technical expertise to evaluate proposals and that communication between the systems developers and users is good; little consideration of non-technical issues, such as job content issues, personnel policies or standardisation; an assumption that change must be a top-down, one-off process; tangible results for clients appear late in the cycle; methodologies based on this approach tend to be paper intensive and bureaucratic; the phases are artificial constructs and in practice are not easily separable and; resulting systems which are difficult to change (Eason 1988, Frenzel 1996, Krogstie 1995, Lewis 1994, Nandakumar 1993, Siddiqi and Shekaran 1996, Truex 1993).

Nandakumar (1993) criticised the SDLC in relation to executive information systems and particularly highlighted problems with the model's conception of time, social context and social process. Activities were often unpredictable, could not be defined into distinct stages and sometimes needed to be completed out of order. The SDLC also did not adequately consider the nature and influence of contextual elements such as structural changes to the organisation and access to users. He also suggested the model did not adequately explain the social processes involved but that systems developers seemed to have a much better understanding of their context than management texts or tools suggest.

The model is widely criticised for not adequately considering the context in which development takes place. Lewis (1994) pointed out the model only indirectly acknowledges changes in the environment in that once one project finishes, another may need to begin. While arguing that the adoption of the lifecycle model was an advance in that it reflected an implicit assumption that the process of development was a *system*, Lewis argued it imported concepts in a generally confused and intermittent manner. The lifecycle model reflects what Kling (1981) termed a discrete-entity model of the context. Siddiqi and Shekaran (1996) suggest that most prescriptive methods, which are based on the lifecycle approach, do not adequately consider the way the context influences the process of producing requirements.

Many criticise the linear approach suggested by the lifecycle model, especially its inability to incorporate pluralism or political rationality and the emergent nature of the planning of change (Quinn 1989; Buchanan and Boddy 1992; Klein and Hirschheim 1993; Walsham 1993). Reflecting strong rationalistic assumptions, there is very little consideration of the micro, social processes outlined in Part 3.

The manner in which the model shows the process unfolding over time is also questioned. Truex (1993) was critical of its top-down nature, stating, "The perspective of those deriving the plan and those left to interpret and implement it are shaped by the hierarchical level within the organisation" (p 59). Lewis (1994) criticises the model for assuming that IS problems can be managed though one-off

projects where, once the project is completed, people go on to sort out some other problem. Any future issue is defined as "maintenance" so that an ongoing need for adaption is underestimated. Thus there is an inadequate understanding of how the actual process unfolds over time.

In essence, there are three fundamental criticisms of the SDLC model. Firstly, there is a lack of consideration of the context. Secondly, it does not adequately reflect the social processes which underlie the process of development at a micro level. Thirdly, there is little acknowledgment of the closely intertwined nature of the planning and implementation of change and the emergent nature of the process, as is discussed further in the next chapter.

4.2.3 Conceptions of process in the LSP

Despite commonly cited criticisms of the model, the SDLC is widely used, and the LSP team were not unusual in employing it. This section focuses on how those involved in the LSP viewed the process of development and compares these conceptions against the dominant SDLC model. This has proved difficult as these perceptions were not always fully and consistently articulated, being continually adapted to suit emerging conditions. In fact, the CIPU's quality management policies actively promoted such adaption, with Jenner (1993) suggesting that,

Processes should be defined by the people who are supposed to be following them. While management and "experts" may initially define the processes they should then be handed into custodianship of their users who will optimise them (p 11).

Nonetheless, throughout the LSP project, the internal systems developers provided a number of concrete examples showing how they conceived the process of development and implementation. This analysis will illustrate that, formally at least, these models largely reflected the SDLC, but that these models did not align well with the actions and concerns of those involved in the project.

During the early stages of the project, the LSP systems developers used the systems development lifecycle as a planning model. The commercial methodology which they used was expressly based on the approach, though it only claimed to provide tools which could be used throughout the lifecycle of systems development, and the LSP team used it in this way. The project management methodology was also based on the lifecycle approach, as is illustrated in Figure 4.2.2. Formally, the LSP team adhered to these models. A review of the project stated the project management methodology was employed to guide team structure, stakeholder identification and management and the management of suppliers while the systems development methodology was used in early and later stages of the project. The detailed design stages, outsourced to the technical systems contractors, were covered by a similar methodology, also based on the SDLC (External review of project 24/2/97). The

project leader's generic conception of the process is illustrated in Figure 4.2.3 and is reflective of the SDLC model.

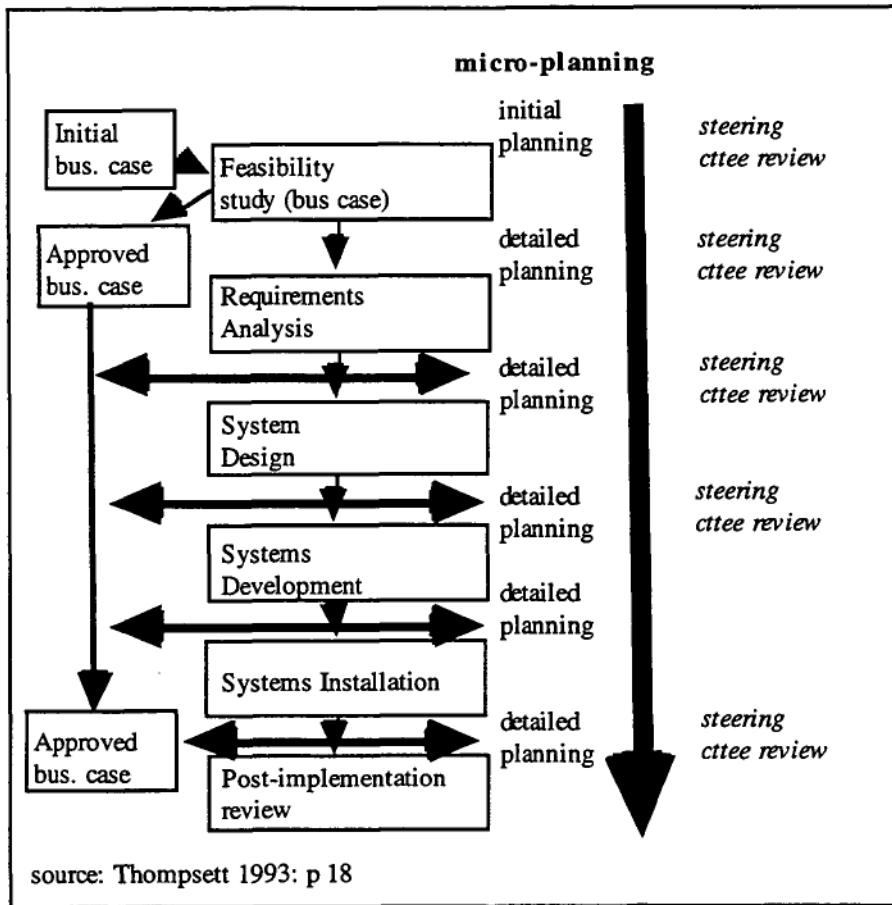


Figure 4.2.2 Model of Project Initiation and Development utilised by LSP team early in the project

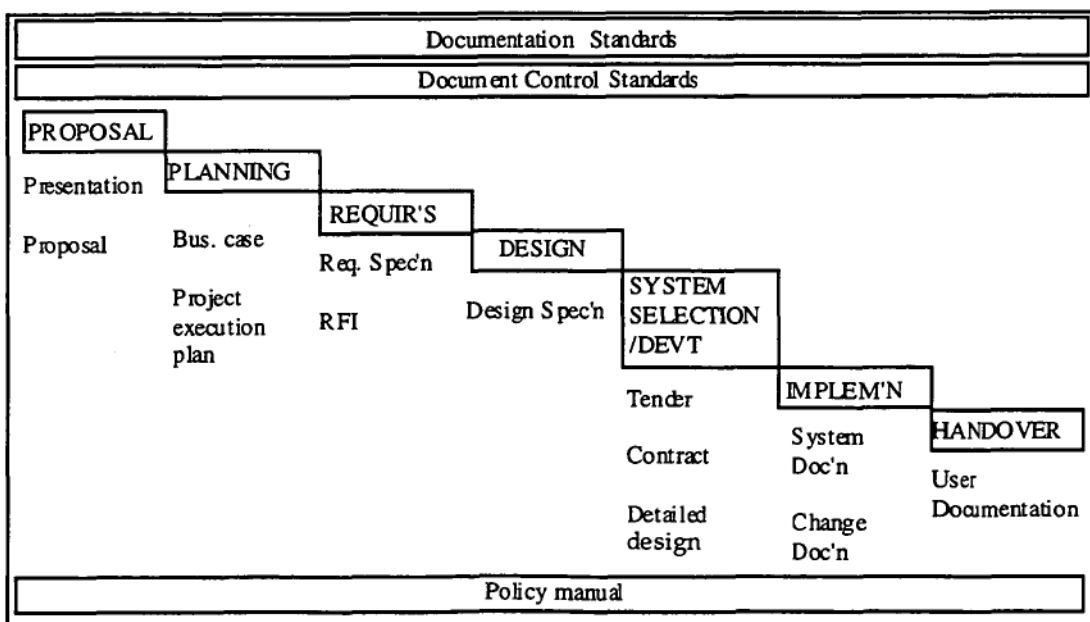


Figure 4.2.3 Generic model of process by CIPU project leader 14/10/94

The first drafts of the business case identified four possible development strategies. These included a phased/monolithic approach, a release/version approach, fast track/prototyping or hybrid approaches and they formally adopted the phased/monolithic approach.

This approach involves the view to developing the system as a whole with each activity standing alone and subsequent activities are not commenced until preceding activities have been completed (Business Case Document v 2.0 14/11/94).

This process was represented diagrammatically as a linear process, but with the associated use of a prototype and a parallel data conversion sub-project. Within each of these sub-projects, development was illustrated as a strictly linear process. Hence the systems developers initially adopted the linear approach of the systems development lifecycle model.

However, although initially adopting an almost pure SDLC approach, it rapidly became corrupted in response to evolving circumstances. Reasonably early in the project, the LSP team recognised that the OPC would have to be gradually introduced to the new technology and they were also unclear what form the system would take. As illustrated in Figure 4.2.4, the implementation of the project was described to the OPC in June 1994 as an emergent process. It was likened to a staircase, where implementation of different aspects of the system was completed at different times "so we don't have to think of too many things at the one time" (14/7/95). Not all tasks were identified at this stage, and the project team openly acknowledged they did not know what they would all be. The step incremental model was not used later, although the intermediate implementation of procedures and technology was an important part of the project, with camera-ready processes and desktop computers being implemented well before the full system, for example. This model indicates the emergent nature of the process and was to better reflect the process of implementing the system than the linear model which formally depicted the development process. Interestingly, though, it was not used later in the project. This may be because it did not align with the CIPU's formal project methodology.

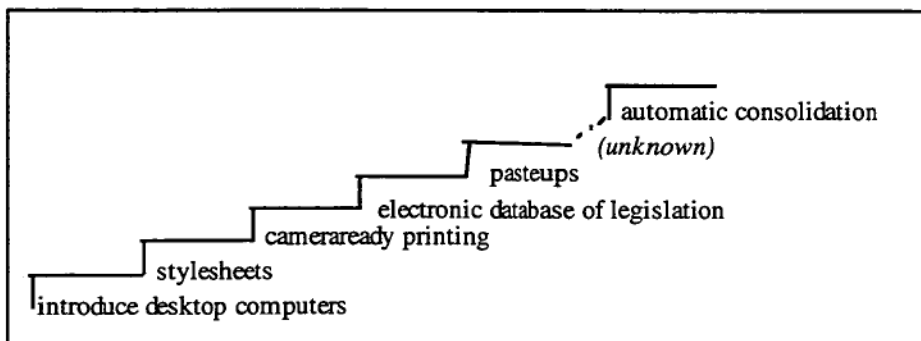


Figure 4.2.4 Description of the LSP implementation process by LSP project team member 14/7/95

Legislation System Project - Phase 1 - 25/11/93 to 16/12/94

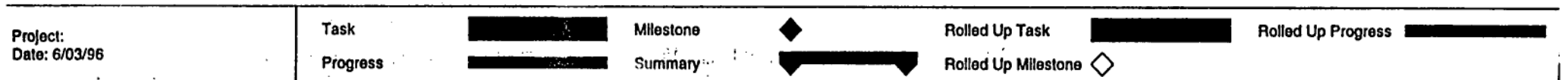
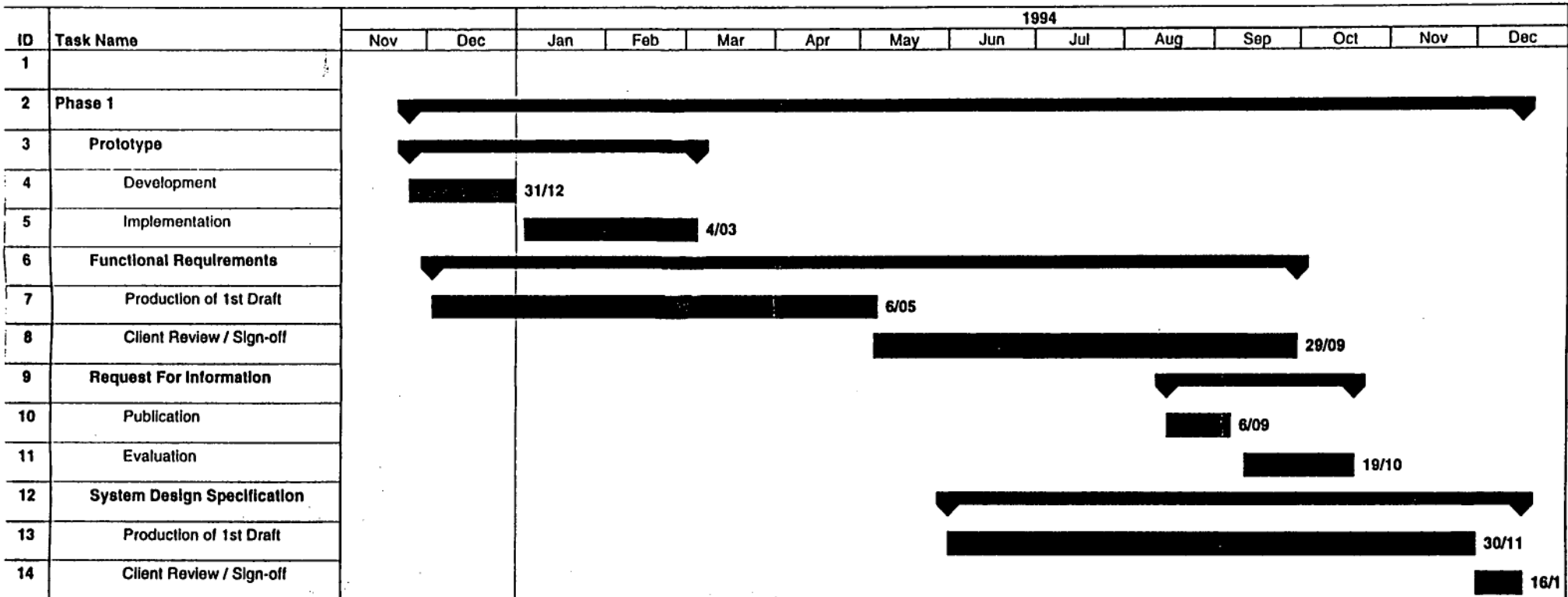
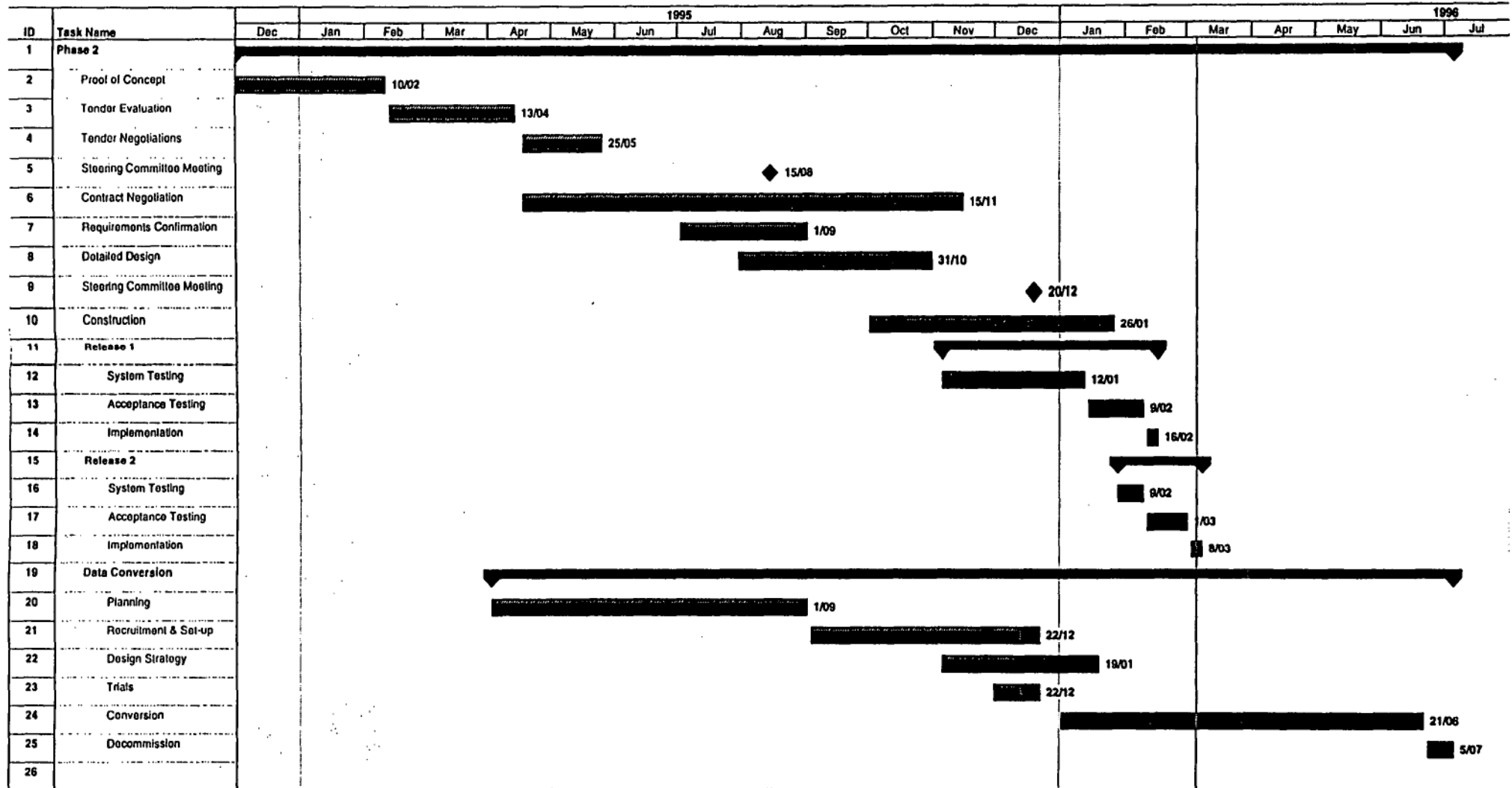


Figure 4.2.6 Early high level LSP Gantt Chart

Legislation System Project - Phase 2 - 1/12/94 to 5/7/96



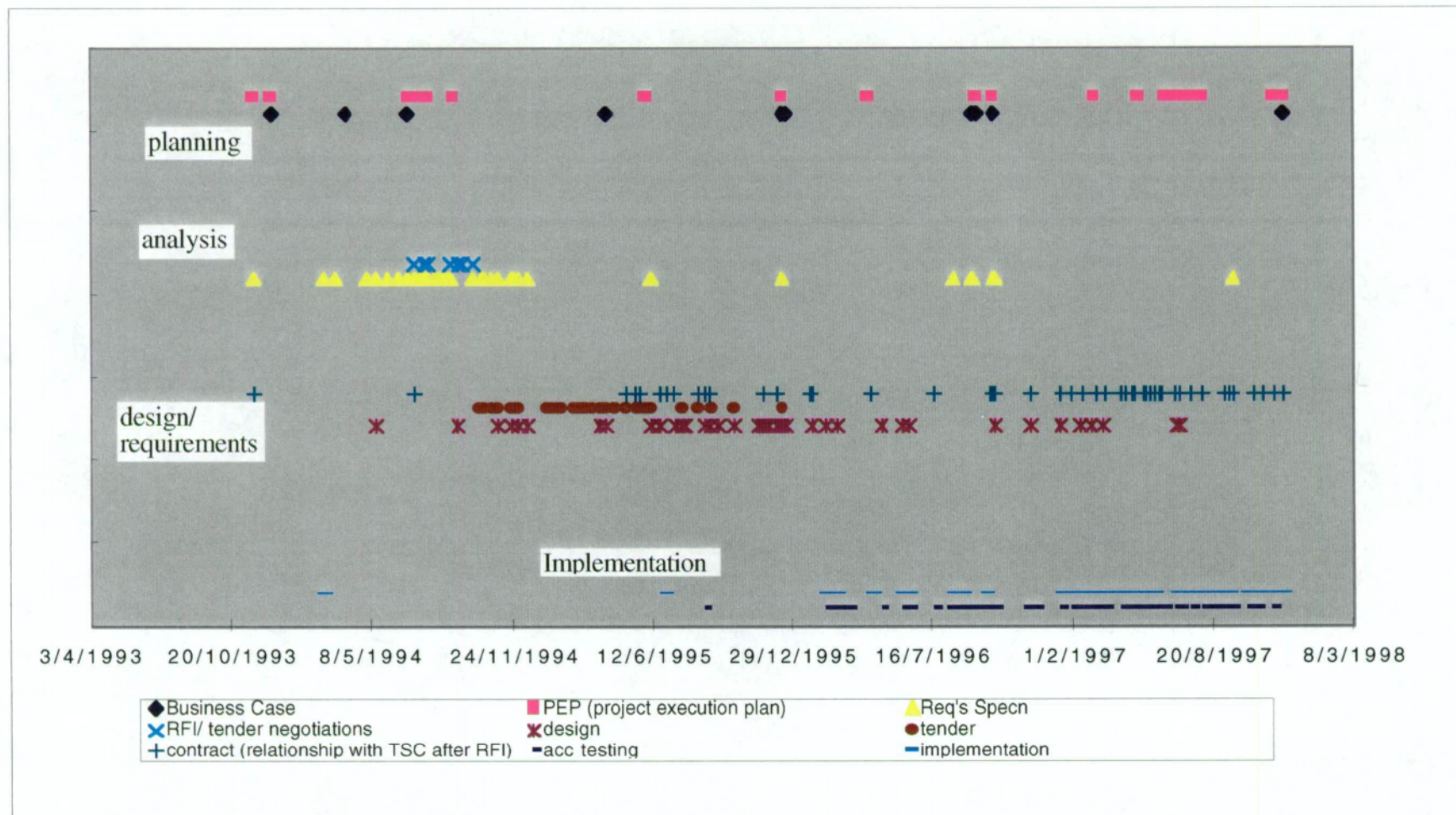


Figure 4.2.7: SDLC related topics discussed in meetings throughout LSP mapped against SDLC model employed (LJ Hocking, "The Systems Development Lifecycle in practice", AIS, Baltimore, August 14-16 1998)

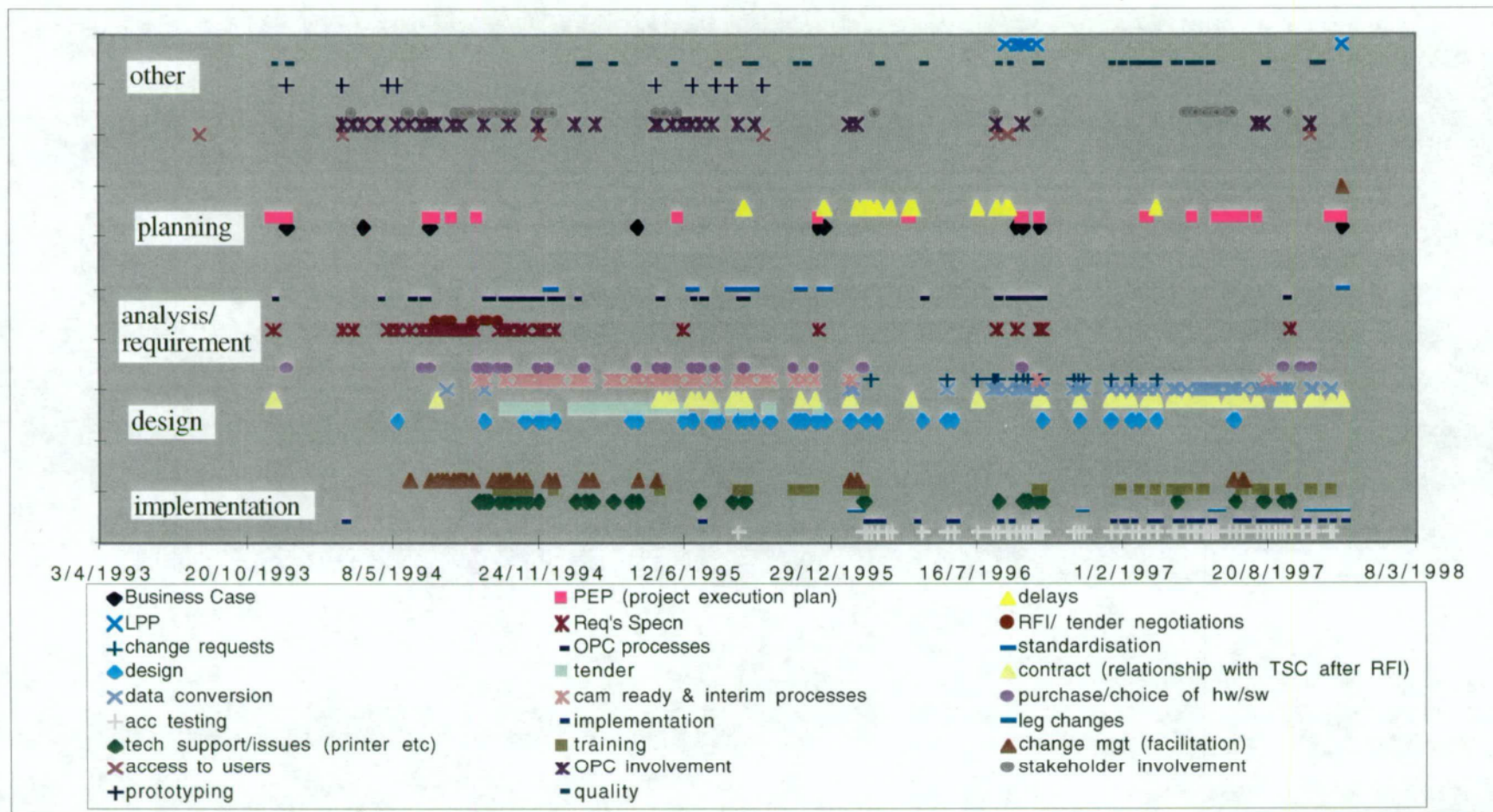


Figure 4.2.8: All topics discussed in meetings throughout LSP mapped against SDLC model employed (LJ Hocking, "The Systems Development Lifecycle in practice", AIS, Baltimore, August 14-16 1998)

While completing the functional requirements document in March 1995, the continued use of a linear process model was increasingly conflicting with the actions of those involved in the project and their requirements. Prototyping was used to aid the definition of requirements and was also used at other times to "make it clear we are talking about the same thing" and "to give a real idea of what it will look like". A proof of concept stage was included when more information about the technical solutions on offer and negotiations with the chosen technical systems contractors took longer than anticipated.

Near the end of the project a further project phase was defined. This phase focused on the utilisation of the system by the OPC and was to be managed by the Outcome Realisation Plan, discussed in Chapter 4.3. Until this phase was defined, the project formally completed when the new system was "rolled" into the OPC and the drafters and their support staff started using it. The inclusion of this phase 3 signalled a recognition that the project could not be considered satisfactorily completed without its effective utilisation by the OPC. Thus, the process emerged over time, but the formal models employed by the systems developers did not reflect this.

4.2.4 Interim procedures and processes

There were a number of interim processes and procedures in the LSP which illustrate the emergent and non-linear nature of the change process in reality. Although not included on public or formal process models, those involved in the LSP spent considerable effort on the development of templates and camera-ready and other interim processes, suggesting the formal models of process did always reflect practices which were more incremental than the models would imply. These interim steps were important in the overall process of development, but were not well acknowledged by the SDLC model formally adopted. Interim processes discussed here include: training; the development of templates and camera-ready processes; the implementation of desktop computers; and the introduction of the drafters to computers.

a) Further training for administrative assistants

While the administrative assistants had a considerable amount of experience with wordprocessing, they had had very little formal training at the beginning of the LSP. Hence they needed training on issues such as swapping between and resizing windows, producing automatic tables of contents and footnotes and the use of styles. Printed legislation required some complicated formatting and the administrative assistants had been formatting the documents on an *ad hoc* basis. They knew the rules for formatting separate parts of the documents and applied them each time they

needed to change. A consultant gave the administrative assistants a course on these issues and introduced the concepts of predefined styles, which the administrative assistants generally found useful. Although not a formal aspect of the LSP, the administrative assistants' skills were important to the project and so the LSP team facilitated training.

b) Development of office styles and templates

The development of office styles formed the basis of templates and were a useful learning tool for analysing requirements for the camera-ready processes and the data type definitions of EnAct. The styles and templates were developed by the remedial trainer, with the assistance of the administrative assistants and the office's executive officer.

The templates included several sophisticated wordprocessing functions such as automatic numbering of pages and sections and automatic cross-referencing. Renumbering sections had been a time-consuming exercise for the administrative assistants as the drafters would often rearrange them in different versions of the draft legislation. The administrative assistants were also impressed with the concept of cross-references that could be automatically updated when they were changed as this was also a time consuming and tedious part of their work. The administrative assistants, however, did not always use this facility, as they were generally under a great deal of pressure to complete work quickly. As cross references were previously part of the drafters' role, some confusion entailed as to who was responsible for them for a period. These procedural and role changes would pre-empt the implementation of the full EnAct system by two years, but were closely associated with the LSP.

c) Camera ready processes

Another interim process was the development of camera ready processes. The term 'camera ready' refers to the final version of a document that requires no modification to its content in order to reproduce it as a publication.

The OPC currently send an electronic document (that may be incomplete, ie. requiring the insertion of running section numbers in page headers, diagrams etc.) to the [printers]. Once received, it is then converted to a different style of document for image insertion and printing purposes.

Under the new camera ready process the final version of the document will be produced within the OPC (ie 100% complete), and then a printed copy of the document will be forwarded to the Printing Authority for publication purposes (discussion paper 27/4/1995).

These camera ready processes were completed by the same consultant who had provided word processing training, with the help of OPC support staff, . Providing an early deliverable for the OPC, the camera ready processes were originally intended to only be used for a short time until the new system was implemented. However, with

the delays in implementing the EnAct system, they were utilised for over two years and can be considered one of the most successful parts of the project.

These processes involved primarily process, rather than technical changes. They simply used existing wordprocessing technologies to produce camera-ready versions of documents rather than sending the content of the legislation to the government printing office for formatting. However, they were considered major process changes in the office and introduced some of the major procedural and role changes associated with the LSP.

Initially, the project manager expressed concern about the way the camera ready processes would fit in with the rest of the project but later suggested two main reasons for it. Firstly, it would help the administrative assistants become familiar with typing up legislation in a structured format and the office used to producing legislation in this way. Secondly, it produced detailed specifications on what legislation looked like to feed into the LSP, the camera ready processes being described as "almost a manual prototype" (transcripts 7/11/94, 21/4/95). Thus the camera ready processes can be considered an interim process in terms of LSP development and implementation.

d) Implementation of desktop computers

Another of the interim stages of the LSP envisaged by the systems developers was the early implementation of desktop computers. By introducing the drafters to desktop computers before the full system, the systems developers hoped to reduce resistance to EnAct itself. This early introduction was also in response to requests from the users who were keen to obtain skills in using the technology. This may be partly because the drafters wished to gain some experience with computers so they would be able to comment on the prototypes the developers presented to them. The administrative assistants also looked forward to obtaining the new computers as the ones they used at the time operated with a small amount of memory.

In August 1994 the OPC were informed they would have new desktop computers by later that year, but did not until late the following year. The developers were concerned that if they bought one type of desktop computer, they might have to be replaced because of the technical solution provided by the tenderer. The main concern was not the cost of the hardware and software, but the time and effort that would be required to train the users in two types of computers. Hence, the purchase of the desktop computers was postponed until a tenderer had been chosen and emerged over time. The desktop computers were introduced for the administrative assistants in August 1995 for the administrative assistants and in September for the drafters.

The systems developers were more wary about introducing the drafters to the technology. They planned to introduce the computers in conjunction with typing, wordprocessing and computer skills courses so the drafters would have the necessary skills to effectively use the technology and not become disheartened. The experience with the first typing course and the drafters reactions to early prototypes had illustrated the need to carefully coordinate training and the introduction to new technology. These courses were being planned for September and November 1995 to fit in with both the demands placed on the OPC during parliamentary sitting times and the planned completion of system (transcripts 25/8/1995).

Some drafters, however, became frustrated. They could not understand why the systems developers were stalling. At this stage the computers were sitting in boxes in the system developers' offices and one of the drafters commented in one of the fortnightly meetings:

We've got to see something physical. I feel we're not getting anywhere. It's been going on like this for months, years... (transcripts 25/8/1995)

The project manager restated the above mentioned reasons for waiting and suggested that perhaps they could compromise by putting computers in the spare office for people to use when they had the time or inclination. As it happened, the desktop computers were in the office for about two years before the implementation of the EnAct system, again illustrating the emergent nature of the process. The drafters used the desktop computers inconsistently during this time, probably because this utilisation was not embedded in the OPC's work processes at this time and the drafters were often more comfortable using pen and paper.

These interim processes and procedures all indicate the system was beginning to be implemented even while it was still being designed. This stands in contrast to the normative, linear conception of the process and suggests it was much more emergent than the SDLC suggests.

4.2.5 Public articulation of process

Even while camera ready and other interim procedures were being developed and implemented, the process was still described in purely linear terms to stakeholders. When giving presentations to stakeholders of the project, the executive manager described the progress of the project in relation the model outlined in Figure 4.2.5. Stars were placed next to the three first boxes to indicate these phased were completed. Hence, publicly, at least, the project was a purely linear one.

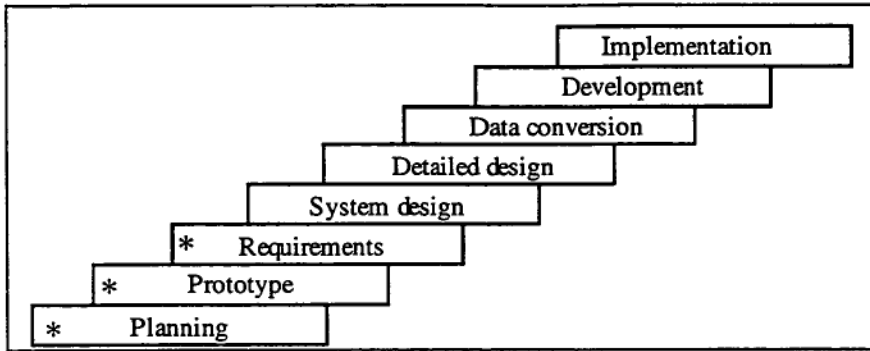


Figure 4.2.5: LSP Project Steps presented to stakeholders 6/9/94

The process by which the LSP was planned to occur is also reflected in the project's Gantt Charts. These charts broadly reflect the SDLC approach but included more overlap between phases. Figures 4.2.6a and 4.2.6b illustrate anticipated timing of tasks early in the project. As the project unfolded, these Gantt charts were changed to reflect evolving circumstances, but largely held the same form, only greatly delayed (see next chapter).

Using only available documentation to gain an insight into how the process of systems development was conceived and unfolded in reality proved problematic. To gain a deeper understanding, topics that were discussed during regular meetings throughout the project were compared with the CIPU's Gantt and other charts. Regular OPC staff meetings, project management meetings, steering committee meetings and project directors meetings were held throughout all or part of the project and the topics discussed at such meetings give some indication of activities at that time. Weekly or fortnightly project status reports also give an indication of activities and concerns throughout the project. While still not a firm indication of the project team's actions, it does give an indication into their focus at that particular time.

The time period covered in this analysis is from mid 1993 to the beginning of 1998, a period of four and a half years. This almost covers the length of the LSP. Figure 4.2.7 maps activities obviously related to phases in the SDLC model outlined in Figure 4.2.3²³. Broadly, it suggests that the SDLC model does reflect the actions of those involved in the project, but that there was even a greater degree of overlap between tasks than even the Gantt charts suggested. For example, the scope of the project was discussed on several occasions while the functional requirements were being developed and it, the business case document and the strategy for developing the system were occasionally raised until just before implementation activities were being discussed.

²³ Here the design and development stages are combined due to the difficulty in differentiating between design and detailed design activities.

Yet these activities form no more than 32% of topics under discussion at these regular meetings. Other issues discussed include: the acceptance of the project or system by the OPC; change management; users' lack of knowledge; training; the development and implementation of interim processes such as camera-ready processes; relationships with the technical systems developers; delays and related projects. Many of these topics can be associated with particular phases of the SDLC model. Planning activities could also include issues of delays, for delays entail that plans have to be changed. Change requests could be considered as part of requirements analysis, as it suggests that requirements are still being determined while "design" could also include the purchase of the technology required. Activities contributing to the implementation of the system might not be always referred to as such, but as "acceptance by the OPC", "change management" or "technical support for the OPC". Other tasks which are difficult to identify with phases of the SDLC include issues of standardisation, quality issues, the development and implementation of interim camera ready processes, prototyping and OPC/ stakeholder involvement. However, their inclusion in Figure 4.2.8, along with the other tasks, gives a better insight into how the process of the LSP actually unfolded.

Particular issues of note include:

- Planning issues remained a consideration throughout the project, as were requirements specification, especially if one includes ongoing change requests;
- Issues concerning stakeholder involvement or access to the OPC remained a concern throughout the project, but were most significant at the time the requirements were being defined and acceptance testing and implementation stages;
- Interim and ongoing processes, such as quality management and the implementation of camera ready processes are not covered by the SDLC model;
- After the initial implementation date of early 1996 was missed, there were fewer issues to discuss generally, but the delays were a concern. Later these concerns about the ongoing delays seem to have been redefined as contractual issues or issues involving the relationship with the technical systems contractors. By this time, the delays had become the norm, and those involved in the project aimed to resolve them and bring the project "back on track". This resulted in the contract being redefined, a new deadline being established and other measures;
- With the ongoing delays, change requests became an increasingly common concern. However, these seem to have been limited as relations with the technical systems developers became an increasing consideration and all involved in the project came under a great deal of pressure to finish it;

- While acceptance testing and implementation activities seem to have occurred in parallel, implementation activities are expected to continue for some time, while acceptance testing has finished.

When we consider all these other issues discussed in regular meetings during the project, the overall process becomes more complex and untidy than either the SDLC and Gantt charts would suggest. Implementation issues are being considered while the system is being planned, even if they are not termed as such. Plans have to change as circumstances demand it and planning is a continual activity. Generally, the division between planning, design and implementation becomes blurred, while the SDLC aims to keep them distinct and separate.

4.2.6 Reflections: An inadequate conception of the process

As discussed above, the lifecycle model is often treated as an objective fact and the idea that there could be alternatives is often not considered. The developers of the LSP were thus not unusual in initially and formally adopting the SDLC model when thinking about the broad process of development.

Yet the model increasingly did not match the reality of how the project unfolded. The confusion in the profusion of models used by the LSP team seems to reflect a balancing act as systems developers adopted the broadly linear models of the systems development literature, but had to adapt it to the local circumstances and recognise the emergent and inherently messy nature of planning and implementing change. That change was emergent is reflected in the many changes to the development strategy, the interim processes and procedures and the number of different models used. The LSP team tried to fit the messy, constantly changing reality into formal linear models and a need to reflect a rationalistic managerial approach. Aiming to adhere to recognised best practice, they adopted standard approaches of the process but were not able to strictly adhere to them and operate effectively in their organisational situation.

Despite its strengths, the lifecycle model still contains some questionable assumptions about the realities of organisational situations and observations of systems development projects such as the LSP suggest a wide gap between espoused theory and theory in use. Westrup (1995) suggests that systems developers can exhibit a much greater degree of organisational awareness than their models and methodologies would imply. Observations of the actions and models of the CIPU supports this stance, specifically in relation to how the overall process of systems development is conceived. The normative literature tends to emphasise that phases of a project need to be distinct, and even that they are strictly sequential. Formally, those involved in the LSP were aligned with these principles broadly, at least to a degree, but when one

looks at the issues discussed at the time, reality emerges as being far more complex and messy. On the surface, observations of how people involved in the LSP viewed the process of development were confusing because they were inconsistent and conflicting. However, when one looks at the actual issues and when they were discussed in relation to the models employed, one starts to suspect that the models employed are not adequately describing what is actually occurring and that those involved in such projects are also having trouble aligning their actions to such models and that an alternative process model which reflects the emergent nature of the process is needed.

In summary, the main criticisms of the systems development lifecycle are that:

- it ignores the dynamic nature of the context;
- it does not adequately consider the social processes involved, such as the political nature of systems development, the issue of user involvement and so forth;
- it assumes a (broadly linear) top-down process of planning and implementation, with a separation between planning and implementation and does not recognise that change is often emergent.

The first two criticisms have been broadly addressed in Parts 2 and 3. Part 2 illustrated the ongoing importance of the context and suggested that changes in it needed to be incorporated into the development process. Formally, the systems development lifecycle model does not recognise this. Requirements are frozen after the requirements determination stage, with all further changes subject to strict change control procedures. The Part 3 illustrated that at a micro, or interpersonal level, the process can be described as negotiation, the use of authority and the application of expertise. It argued that, even when this was not formally recognised in the systems development literature, such socially dynamic processes were obvious throughout the process of development. In other words, the commonly used SDLC model inadequately reflects the ongoing impact of the organisational context, the social interaction which make up the systems development process at a micro level and the closely intertwined nature of the planning and implementation of such changes. Table 4.2.1 provides some examples of these criticisms in relation to the LSP.

Dynamic nature of context	Social processes	Inadequate conception of process
(addressed broadly in Part 2) <ul style="list-style-type: none"> • had to fit in with OPC's workload • possibility of changes with new government • user frustration at inaction • the possible effect of enterprise bargaining on the LSP • the effect of plain English drafting • "wish list" of changes (see below) • effect of delays (see below) 	(addressed broadly in Part 3) <ul style="list-style-type: none"> • CIPU recognition that user acceptance of system was crucial • recognised need to obtain broad commitment to project • reaction of users to prototype and functional requirements • active role played by drafter developing prototype • OPC involvement in designing system (eg editor evaluation exercise) • creation of standards (see below) • quality review procedures (see below) 	Non-linearity of process <ul style="list-style-type: none"> • Camera ready and other interim processes • changes to business case throughout the project • delays (see below) • need to change decision made in editor evaluation exercise due to further information • use of prototypes • emergent issues (eg paper size) • actions not in chronological sequence (eg Technical systems contractors began designing before contract signed) More examples discussed below

Table 4.2.1: General criticisms of the SDLC model

A lack of consideration of the context of systems development and the socially dynamic nature of the process at a macro level are at least partly attributable to an inadequate conception of the process as embodied in the systems development lifecycle model. The context of the development process is only really considered in the planning and requirements determination stage. Once the functional requirements document is "signed off", the context in which the system will be used is rarely considered until the implementation stage, when implementation problems are often attributed to issues of change management. As discussed in Chapter 4.1, the concept of change management is often a "string and sticky-tape" or "rough and ready" solution to fundamental problems in how we view the process of systems development. The systems development lifecycle approach tends to ignore the socially dynamic nature of change. It adopts an objectivist ontological stance where problems are seen to exist independently of whether we perceive them or not and need to be analysed after the "facts" have been gathered. These issues have been addressed at length in the preceding chapters.

The focus of Part 4 is on the third criticism: the top-down nature of the process with its implicit separation between planning and implementation activities. The lifecycle approach is sometimes referred to as the "waterfall" model. Iterations are recognised, in much the same way that salmon ladders are built into dams. Change flows like water down a river, from step to step, from one deliverable to another. Iterations are movements against the flow of change in which we usually drift, but are necessary in

order to navigate the right path. The flow down the river is illustrated as a series of arrows from one stage to the next, but how and why we move between them is rarely examined. The systems development lifecycle illustrates change as a series of tasks to be completed in a particular sequence. Using the metaphor of the river, it focuses on aspects of the route – dams, waterfalls and so forth – rather than the actual process of change embodied in the moving water. This dissertation aims to produce such a model of process and consider how it is reflected in alternative normative models.

In summary, this chapter has:

- described the commonly used systems development lifecycle model of the process of systems development and implementation;
- outlined criticisms of this model made by others;
- argued that, while the systems development lifecycle is intended to be a model of the process over time, it does not adequately address the dynamic nature of context, the ongoing influence of social processes and the often non-linear nature of change; and
- illustrated that, even though the SDLC model was formally employed in the LSP, the actions of those involved did not greatly align with the model. Particularly, mapping topics discussed in ongoing meetings against the formally employed model suggests a low correlation between the two. In other words, even where the model is used in practice, it does not reflect the activities of those involved very well.

Chapters 4.3 and 4.4 suggest an alternative macro model which does not suffer from these criticisms.

4.3 Chronological and hierarchical emergence

The relationship between planning and implementation change is a key issue when considering the process of change. This relationship should be viewed as emergent both chronologically and hierarchically. Chronological and hierarchical emergence is illustrated through a description and analysis of two minor case studies, HRMIS and REMUS and the final stages of the LSP.

This chapter completes descriptions of the LSP and critiques of normative systems development practices. Based on observations and arguments from previous chapters, it sets the scene for the following chapter which identifies alternative models of process at the dasien level of temporality. While this is a long and complex chapter, the points made here are pivotal to the main arguments of this dissertation.

The relationship between planning, development and implementation is a key one when considering the process of information systems development. The systems development lifecycle model presents these components of the change process as relatively sequential activities. Most significantly, there is an implicit assumption that there is a division between planning and implementation, even when iterations are recognised. This is well-illustrated through Avison and Fitzgerald's' (1995) comparison of methodologies, where more than half of the eleven methodologies evaluated go no further than designing the system (p 465). The programming, testing and implementation of the system is examined by only a few methodologies, suggesting that most are based on an assumption that the implementation of plans simply unfold once planned and designed.

Here these assumptions are questioned and, through a consideration policy analysis literature, an alternative model is suggested which reflects both hierarchical and chronological emergence, rather than the top-down process dominant in the systems development literature. These concepts form a basis for evaluating alternative normative models of the systems development process in the next chapter.

4.3.1 The relationship between planning and implementation in the policy analysis literature

The policy analysis literature is of great relevance as systems development is basically policy implementation. Significantly, policy analysis literature focuses on the way that policies are implemented, the relationship between the planning process and the manner in which these plans become operational. The focus on implementation is a response to a perceived over-concentration on policy planning and literature. This stands in contrast to the planning focus of most information

systems development literature, where implementation is relegated to simply "installation" or conversion (for example Shelly, Cashman et al. 1995). Yet implementation is the crux of the whole process. Anyone can plan. I can make plans for the UN in Bosnia, but I doubt they would be implemented. It is the realisation of these plans which is the crucial issue. The policy analysis literature focuses on these issues.

Three broad approaches to policy implementation are recognised: top-down/forward mapping, bottom-up/backward mapping and emergent (Younis and Davidson 1990). The difference between the three centres on the relationship between planning and implementation. Here it will be argued that, while only the last has descriptive validity, most normative systems development literature, including that based on the systems development lifecycle, assumes a top-down approach.

a) The top-down approach and problems with it

A top-down approach assumes that policy is initiated at the top of the organisation and is later implemented by others further down in the hierarchy. This perspective is aligned with rationalistic management assumptions which assume that members of an organisation are guided by common goals and management controls the actions of others. It also aligns with the second definition of change management described in Chapter 4.1 (Keen 1981; Mintzberg 1993).

The top-down approach is criticised for assuming the relationship between planning and implementation is purely a sequential one. That is, action follows on from clearly specified plans. Writers, such as Majone and Wildavsky (1978) suggest that the top-down model is unrealistic as many constraints are hidden at the planning stage and are not uncovered until implementation. Pointing out the objectives of most policies are multiple, conflicting and vague and that the people making them cannot possibly be omniscient, they comment,

Literal interpretation is literally impossible unless a policy is narrow and uninteresting... then policy will never be able to contain its own consequences. Implementation will always be evolutionary, it will inevitably reformulate as well as carry out policy (Majone and Wildavsky 1978: p 116).

That is, implementation really occurs in an emergent manner, both in terms of the organisational hierarchy, and in terms of time.

Elmore (1979) referred to top-down policy implementation as "forward mapping" and described it as the "strategy that most readily comes to mind when one thinks about how a policy-maker might try and affect the implementation process" (p 603). He commented most management science and decision analysis literature reflects forward mapping assumptions and that many management tools, such as the critical path method and PERT diagrams also build on such assumptions. Formal strategic

planning can also be included in this paradigm as it implicitly assumes that planning and decision-making can be separated from their implementation, as can project management methods. These models are widely used in the systems development process.

Elmore's (1979) article is primarily a criticism of the forward mapping (or top-down) approach. He believed that while forward mapping is used prescriptively, it does not adequately describe what happens in organisations.

The notion that policy-makers exercise- or ought to exercise- some kind of direct and determinant control over policy implementation might be called the "noble lie" of conventional public administration and policy analysis... (Elmore, 1979: p 603).

Forward mapping, he continued, assumes that implementation will be improved by more explicit policy directives, better attention to administrative responsibilities and clearer statements of intended action. In pursuing these actions, the paradigm reinforces what Elmore calls the "myth" that implementation is controlled from the top. He also criticised this approach for only considering a narrow range of reasons for implementation failure. It only produces standardised solutions which he believed are "notoriously unreliable" (p 610) and difficult to adapt and does not consider the benefits of discretion when implementing policy. Thus, a top-down approach is an inadequate explanation of the process of planning and implementing change.

However, such rationalistic assumptions have dominated the IS literature (Yetton, Craig et al. 1995). As discussed in Chapter 3.1, IS planning reflects a strong top-down approach, if one accepts the texts in the area. Ward and Griffiths (1996) suggested some of the desirable characteristics of an IS/IT planning approach including a need for a top-down, consistent view of the whole area, with clear objectives and principles for application development. Reflecting strong rationalistic assumptions, they comment,

...any strategic planning approach should provide management with a vehicle to make rational decisions. These decisions should be made at logical and clearly defined checkpoints, which break the whole process up into easily comprehended units of work, and prevent wasted time on unwanted deliverables (p 131).

Korac-Boisvert and Kouzmin (1995a 1995b) suggest that the adoption of the systems development lifecycle through the use of methodologies reflects rationalist decision-making and a sequential, top-down approach to the process. Looking at a wide range of methodologies, they found that most still assumed a top down process and human actors who were rational decision-makers capable of cognitive perfection. They argued most methodologies reduce detail to a simplistic level and restrict informal contact between participants in the process. Commenting this is inconsistent with an increasing number of environmental changes, they suggest there is a gap between "espoused theory" and "theory in use". This study backs up their conclusions that

systems development tends to reflect an inadequate top-down conception of the process.

An implication of a top-down approach is that requirements can be "determined" entirely at the beginning of the project, yet this is simply not possible. As Jones and Walsham (1992) effectively argue, there are limits on the knowledge that can and should be obtained during the development process. This is due to a number of reasons. Firstly there are theoretical limits. What we know is culturally conditioned, contingent on the context and based upon the language employed. Knowledge is socially defined and so is derived from an ongoing process of negotiation. Hence design knowledge will be emergent and intersubjective, highly influenced by the social processes and frames of reference of the design team. There are also practical limits. Organisations are not stable entities so we can never fully understand them. Some aspects will only be observable to people with a deep understanding of the situation, such as participants or participant observers, yet this is often not practical for system designers. There are also ethical issues to consider. Jones and Walsham point out that systems development methods and tools used to support the process generally assume that developers can obtain a complete understanding of the situation for which an information system is being designed. However, in many cases, this is simply not feasible.

If we accept that knowledge is partial, contingent and subjective, then there are important implications for the development process. Siddiqi and Shekaran (1996) comment that while most practitioners recognise implementation may proceed from incomplete requirements, much normative literature and research does not. A survey of the normative literature of systems development backs up these comments. Most recent texts assume at least a partially top-down approach to development, with half the texts surveyed assuming that a strictly top-down approach was in order (see Appendix 1). Those which assumed a partially top-down approach recognised that there would be some iterations within the structured process, or that initially feasible projects may later become infeasible.

Information systems development and implementation is innovation, therefore designs can never be precisely predicted (Sauer 1993). Simply put, the assumption that systems development can be an entirely planned process is questionable.

b) The bottom-up approach

Elmore introduced the idea of backward mapping as a possible alternative to the top-down approach. The logic of this "bottom-up" approach to policy implementation is opposite to that of the top-down, forward mapping approach (Younis and Davidson 1990). Elmore described it as beginning,

...not at the top of the implementation process but at... the point where administrative actions intersect private choices. It begins not with a statement of intent, but with a statement of the specific behaviour at the lowest level of the implementation process that generates the need for a policy. Only after that behaviour is described does the analysis presume to state an objective; the objective is first stated as a set of organisational operations and then as a set of effects, or outcomes, that will result from these operations' (Elmore 1979: p 604).

In essence, the backward mapping approach tackles the issues of implementation before and while policy is being created and it is only at the final stage that the policy-maker makes a policy which directs resources.

Unlike forward mapping, backward mapping does not rely on compliance with the policy-makers' intent as the standard for success (p 604). Forward mapping assumes that the implementation process is linked in an hierarchical relationship, with the source of policy being the area with most authority and influence. Success is achieved with clear lines of authority and control. In contrast to this, backward mapping assumes that it is easier to influence policy the closer one is to the problem. Implementation success is not seen to be a result of control and authority, but discretion at the face of the problem.

Elmore states that applying forward and backward mapping to the same problem gives different results. The former tends to emphasise centralised control and factors that are easier to manipulate by policy-makers, such as funding, organisational structures, authority relationships and regulations. Backward mapping tends to stress the dispersal of control with related factors such as incentive structures, the ability and knowledge of people near the problem, the bargaining relations among different levels of the organisations and the strategic use of funds. The former sees policy implementation as an hierarchically ordered process; the other, as a decentralised one (p 605).

Mintzberg's (1995) grassroots model of strategy formation is also a bottom-up approach. It assumes that strategy formulation can be over-managed and that sometimes it is important to merely let patterns emerge rather than force artificial consistency. These strategies can emerge from anywhere in the organisation and become collective when they are diffused and affect the behaviour of the organisation as a whole. This process of proliferation may be conscious, but not necessarily so. Managing this process does not involve preconceiving strategies, but recognising their emergence and intervening when appropriate.

Yet there are problems with this approach (Younis and Davidson 1990, Hocking 1995). It can be criticised for promoting short term solutions or solutions which only meet the needs of the people involved, rather than the whole organisation. It also assumes that those at the "pitface" of the problems faced by the organisation have

similar long-term goals as those at the centre of the organisation and ignores the influence of authority.

c) An alternative emergent approach

Generally speaking, neither a purely top-down or bottom-up approach is appropriate. Younis and Davidson (1990) comment that, while the three approaches to policy analysis of top-down, bottom-up and emergent, are all recommended prescriptively, only the emergent strategy appears to have descriptive validity.

So what are the characteristics of such an emergent model? After criticising what they termed the planning as control and planning as social interaction models, Majone and Wildavsky proposed an evolutionary model of policy implementation which they claim does not suffer from the criticisms they used to dismiss the other approaches.

This model recognises that:

- in most cases goals tended to be multiple, conflicting and vague and constraints tend to be unclear due to cognitive limitations and the dynamic nature of the context. Policies can thus be no more than a cluster of potentialities and implementation begins as possible dispositions for treating certain situations in certain ways;
- policies, even when drafted into laws, only exist as potentialities and their realisation does not simply flow on from the policies, but depends on the intrinsic qualities of the policies and contextual circumstances. The same can be said of plans, and systems development planning in particular;
- implementation can thus shape policy, as initially paradoxical this sounds. Policies are continually changed by their realisation through implementation activities which simultaneously alter resources and objectives. The multiplicity of objectives and constraints such as funding, time, values and conventions means that implementors are faced with an inevitable juggling act as they try and transpose (usually) vague proposals into a concrete reality. Alternatively, the removal of some constraints can provide implementors with possibilities policy planners did not envisage or even perhaps desire. "Unless one is willing to assume that policies spring fully armed from the forehead of an omniscient policy-maker, discretion is both inevitable and necessary" (Majone and Wildavsky 1978: p 113);
- policies may be created retrospectively to provide justifications for actions. As Mintzberg would say, "Planning under difficult circumstances may be better conceived as the interpreter of action, rather than its driver" (1994: p 293);
- people implementing policies act within a context of expectations. Policies define the arena for action, the identity and role of principle actors, the range of tools for

action and the resources available. Most significantly, policy provides a conceptualisation of the problem area to be addressed; and

- when problems are defined through the process of attempting to create acceptable solutions, then implementors help to create policy. Problems may be best understood through solutions and, if this is so, then implementation is not just about finding answers but also about framing questions.

Policy planning and implementation is thus a constantly evolving process. Unless it concerns a very narrow topic, policy or plans can never "contain its own consequences" and implementation will "inevitably reformulate as well as carry out policy". An emergent approach is aligned with Mintzberg and Waters' idea of "walking on two feet, one planned, the other emergent" and Quinn's concept of logical incrementalism (Quinn 1989) but not a great deal of the literature on systems development.

As Mintzberg stated, people are more productive and more satisfied when realising their own plans rather than those of other people. This is because,

- (1) sense of accomplishment is less when executing someone else's plan; (2) there is less tendency to try and confirm the validity of another's plan by executing it successfully - less confidence that it can be done; (3) there is less commitment to see that the plan works well; (4) there is less flexibility and less room for modification and initiative to make improvements to an assigned plan; (5) there is less understanding of an assigned plan; (6) human resources are not so well utilized; (7) there are more communication problems and consequent errors and distortions in following instructions; (8) there are competitive feelings aroused between planners and doers, to such an extent that it appears that if the former "win" the latter "lose" (Bass 1970 in Mintzberg 1994: 164)

Due to the socially dynamic nature of the process, systems development, like policy planning, needs to be emergent. Truex (1993) suggested if we view organisations as emergent entities, information systems had to be designed incrementally rather than in the rational comprehensive way they are generally. Pinto and Prescott (1990) describe the "mutual adaption" which occurs between project managers and users through attempts to meet client needs and comment that planning must be continually reassessed in the light of client demands. Thus the concept of negotiation is crucial in understanding the emergent nature of the systems development process.

Orlikowski and Hofman's (1997) improvisational model of change management is a recent example of a chronologically emergent model. Echoing work by Barley and Tolbert (1997), they suggest that often the implementation of new technologies involves anticipated changes that can be planned, emergent changes which arise from local innovation and that are not originally intended (hierarchically bottom-up – see below) and opportunity-base changes, which are not anticipated ahead of time, but which provide unexpected opportunity (chronological bottom-up - see below). These three types of change build on each other. They illustrate how these different kinds of changes were effectively combined when implementing groupware technologies:

We propose that people's assumptions about technology-based change and the way it is supposed to happen are based on models that are no longer appropriate. Traditional models for managing technology-based change treat change as a sequential series of predefined steps that are bounded within a specified time (p 20).

They liken approaches to managing change to methods of navigation:

The European navigator begins with a plan – a course –which he has charted according to certain universal principles, and he carries out his voyage by relating every move to that plan. His efforts throughout his voyage is directed to remaining 'on course'. If unexpected events occur, he must first alter the plan, then respond accordingly (p 11).

On the other hand, the Trukese navigator begins with an objective rather than a plan:

He sets off towards the objective and responds to conditions as they arise in an ad hoc fashion. He utilises information provided by the wind, the waves, the tide and current, the fauna, the stars, the clouds, the sound of water on the side of the boat, and steers accordingly. His effort is directed to doing whatever is necessary to reach the objective (Berreman 1996, in Orlikowski and Hofman 1997: p 11).

They suggest most people think about managing change in the same way the European approaches navigation, starting with a plan to which future actions are aligned. They liken the alternative improvisational model of change management to the Trukese approach to navigation. The corresponding approach to change recognises that, while change can be sometimes anticipated, it can often be emergent or opportunity based. They provide a case study illustrating how this improvisational model unfolded with the implementation of groupware technologies in a large software company.

The primary difference between the European and Trukese approach to navigation is that the latter refers to navigation by sail while the former refers to navigation by steam, or some other mechanical means which do not rely on the prevailing winds to the same degree.

...On the desk the ship's log lay open at the page of entries for her last day at sea.

"Barque *Rothsay Bay*. Auckland towards Sydney. 53 days" it read.

"Auckland *towards* Sydney? Why not *to* Sydney, sir?" I asked.

"A sailing-ship isn't bound *to* any place until she gets there," said Jackie in a loud voice...

"Steamers think they are bound to places," he went on. "Sailing ships know they are always bound towards. Towards! That's the thing. Always towards - and you do your damnest to get there. But you aren't sure of anything until you arrive..." (Villiers 1955: p 57-8)

The key issue with the emergent approach to change is that one recognises there are likely to be influences on its process and content which are just as uncontrollable as the winds at sea. Plans can be made for achieving particular changes, but our achievement of those plans is dependent on a large number of issues, only some of which can be planned and controlled. In other words, we can only plan to move *towards* a particular change objective, not to it. Any adequate model of the systems development process must recognise this.

4.3.2 Chronological and hierarchical emergence

To pursue the points just made, it is worth first outlining a significant issue which arises when considering emergent processes. There seems to be some confusion in the literature as to what is exactly meant by the terms "top-down" and "bottom-up". Sometimes the terms refer to decision making over time, with some assuming that decisions are made at the top, or beginning of the process. At other times the terms refer to the decision making process within the organisational hierarchy. It is often unclear if writers are referring to one or the other or both interpretations. In other words, sometimes writers seem to refer to chronological emergence while at other times they refer to hierarchical emergence. A top-down hierarchical perspective would focus on issues such as authority and control while a top-down chronological perspective would assume that plans are created up-front and their implementation can be later judged according to these plans. A bottom-up hierarchical perspective would focus on the social processes of negotiation surrounding the planning and implementation process while a bottom-up chronological perspective would view plans as only recording or justifying consensual decisions or actions. Figure 4.3.1 illustrates these concepts diagrammatically.

Logically there is a connection between the chronological and hierarchical dimensions, as a greater focus on social processes suggests that changes will be more emergent. Yet the link is not necessarily tight. As Mintzberg (1994) commented, planning reinforces the idea of a unitary central authority. It would be difficult to anticipate management's directives, so the distinction between chronological and hierarchical processes is not important when a top-down process is assumed. However, if the process is viewed as an emergent one, a lack of distinction between the two types of emergence can lead to confusion. Plans can be heavily negotiated up-front and then treated as an objective fact as they are later implemented.

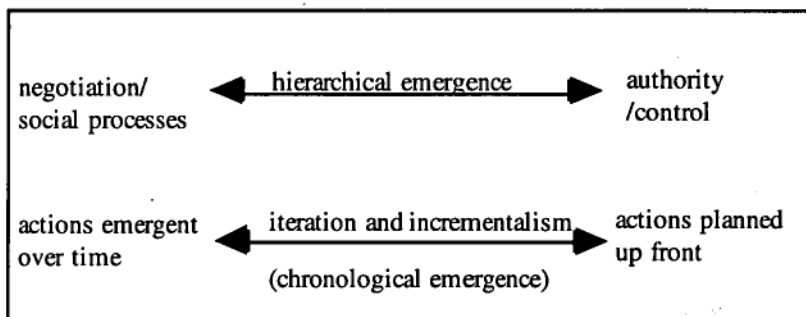


Figure 4.3.1: Chronological and hierarchical emergence

In practice, it is difficult to separate hierarchical and chronological emergence, but conceptually it is important to for clarity. Models of the systems development process generally support iteration, yet rarely acknowledge hierarchical emergence.

A top down process generally assumes that plans will be made by management and will be implemented by others lower in the organisational hierarchy at a later time. Basically, an emergent process model should exhibit both chronological and hierarchical emergence, but it is useful to conceptually separate them.

4.3.3 Chronological and hierarchical emergence in practice

To illustrate what is meant by chronological and hierarchical emergence, it is worth giving an example of what it might look like in practice, before considering it in relation to the LSP and the systems development literature. This is done by referring to two human resource management systems development projects in the Tasmanian State Service. REMUS is an almost service-wide human resource management information systems developed in the Tasmanian State Service by the Payroll/Personnel Consortium from 1992 to the present. Immediately prior to the development of this project, another with a similar scope, called HRMIS, had been attempted but was abandoned. One of the key differences between the two projects was that, while the HRMIS project exhibited only minor hierarchical emergence and chronological emergence only when delays or unexpected circumstances forced it, the REMUS project formally acknowledged both hierarchical and chronological emergence at the highest level. These minor case studies were investigated retrospectively through interviews with people involved and an examination of project documentation. As with the major LSP case study, the descriptions collated below have been reviewed by participants.

a) HRMIS

In August 1988 a project to implement a Tasmanian state government-wide integrated payroll-personnel system was initiated to replace existing inadequate systems for most State Service employees. The user specifications were completed by February 1989 and, after an evaluation of commercially available packages, it was decided to develop a prototype system using an innovative product. This prototype, termed HRMIS, was a joint development project between the Department of Premier and Cabinet (DPAC), the government's centralised computing centre and a commercial organisation. A central development team was established within DPAC.

A prototype system was completed by February 1990 and was transferred to another host for further evaluation before deciding whether to implement the system, the Auditor General reporting that

I am informed that the decision to develop the prototype does not imply an irreversible commitment to in-house development of the system. A final decision will depend on agency acceptance of the prototype (Audit Office 1989).

While the pilot version of the project had been satisfactory, the software did not interact well with the new hardware host. The problem was further exacerbated by a lack of skills in the area, causing the project's schedule to slip considerably and the Audit Office was asked to evaluate the system.

However, due to the government's redundancy and restructuring program, the lack of a service-wide personnel system was noticeable. In December 1990, Cabinet decided to implement HRMIS in all State Service agencies. The Auditor General expressed concern due to the ongoing technical problems and the original plan for agencies to evaluate the system before a decision was made on the system's implementation. The Secretary of DPAC responded the project would not be completed "in undue haste" if Cabinet's July 1st deadline was to be met (Audit Office 1991).

Reviewing the system in December 1992, the Auditor General concluded that the decision to fully implement the limited pilot project "did not fully consider the risks of developing such a large and complex system". This lack of consideration was reflected in the resources allocated to the project (Audit Office 1992). In January 1992, agencies were informed of the plan to involve six agencies in a pilot implementation. They were consulted in February that year but following a quality assurance review, the target date of 1st July 1992 was considered unachievable. In July 1992 Cabinet decided to "...rescind the previous decision that HRMIS be adopted as the service wide human resource management information system..." (letter from the Secretary of DPAC to the Auditor General, quoted in Audit Office 1992). The project was effectively abandoned despite considerable resource investment²⁴.

b) REMUS

Following the demise of the HRMIS project, Tasmanian State Service agencies still had a need for a human resource management system. In late 1992 four agencies formed a consortium to develop user requirements and issue a request for information. This consortium grew to include nine of the twelve agencies in the Service and has recently implemented a service-wide compatible system in those agencies. The steering committee included the heads of each agency in the consortium and reported to Cabinet and the departments who use the system were not only consulted during the project, but ran it. This approach provided the opportunity to pool expertise and achieve cost-savings through economies of scale. It also had the added advantage of producing compatible integrated pay/personnel systems across most of the state service. The Departments who were not included in the Consortium included the

²⁴ Part of this description was included in Hocking, L. *A history of information resource management in the Tasmanian State Service* (honours thesis) University of Tasmania, 1993

three largest agencies of Health, Education and Treasury, but at least one of these is currently considering switching to REMUS. Essential features of the new system were that it had to be a true, fully integrated relational database system capable of meeting a range of diverse requirements in the separate agencies comprising the consortium.

The progression of the project depended explicitly on the acceptance of people from each of the agencies as well as tight scrutiny by the Audit Office. The request for proposals was issued in September 1993 and the selection process continued from then until May the following year. These activities were coordinated by the central project team but also involved agency personnel. The project team primarily consisted of HR people seconded from the consortium agencies and technical expertise was obtained from a reference group of departmental IT personnel and consultants. The project was funded by a levy with departments able to withdraw or join the consortium after paying a fee. It is important to note that firstly, the project was managed by human resource rather than IT people and that, secondly, it was controlled by the agencies who would use the system, not a central agency. This stands in contrast to the HRMIS case study.

In October 1994, the Consortium agencies agreed to a Memorandum of Understanding which explicitly defined the role of the consortium and the responsibilities of all participants. This represented a "constitution" for the consortium, outlining the purpose of the project, the background, rationale, objectives, project scope, a proposed implementation plan, governance structure, project management and teams, reference groups and working parties, budget and funding, terms and conditions of operation and specific agency issues. They then concentrated on acceptance testing, designing data conversion strategies and other pre-implementation issues while awaiting approval for funding.

Given the government's experience with HRMIS and the problems experienced with previous payroll systems, politically this system could not afford to fail. Human resource management systems generally have a bad record due to the complexity of data involved. Salmon and Proud (1993) estimated that only 5-15% of large human resource management systems purchased in Australia had been successfully implemented, and cited the REMUS project as one of the few success stories. Here it will be illustrated that the success of the project can be linked to the fact that chronological and hierarchical emergence were explicitly acknowledged throughout the process.

The consortium was headed by a steering committee made up heads of the agencies involved with the consortium or their representative. A central project team was

responsible for areas of common concern and coordination and was a focus of pooled resources. The implementation of the system in each agency was managed by a project team within each agency, with some assistance from the central team. The Memorandum of Understanding stated,

Ideally each agency will have a project team which is responsible for the implementation of the system within their agency. The project team is responsible to a project manager within the agency. A close cooperative relationship between this team and the Consortium project team is anticipated (p 17).

The central project team consisted of people seconded from consortium agencies and others recruited because of their experience or expertise in human resource systems. At the time of implementation, it included seven people, the maximum number throughout the project. Most of the seven had human resource management backgrounds, with only one person having an IT background. The central team project manager considered himself an HR expert rather than an IT person and believed the fact the system was developed by HR people was a contributing factor to its success as it helped align the systems goals and plans with the business objectives of the agencies. It was also useful to have people with experience of the different agencies in the consortium in the central team, as they were able to bring up pertinent issues. That is, hierarchical emergence was supported by including human resource personnel from the agencies belonging to the consortium.

From May 1994, the focus of the project turned to implementation issues, such as the conversion of data, the acquisition and/or adaption of hardware and software, training and workflow design. Many of these tasks were the responsibility of the agencies and were reviewed by the audit department, the central project team manager and external consultants. Before the system was fully implemented in each department, there were strict testing procedures, including at least two parallel payruns with all important data being checked against the previous system. By late 1995 the system had been implemented in all the agencies.

These time-frames have varied according to the agency. Due primarily to differing requirements, one agency decided to implement separately from the consortium, but used the same system plus the expertise and experience of the central project team. Implementation progressed differently in separate agencies. For example:

- Some agencies used the implementation of the system to promote decentralisation of HR functions (eg Tourism). Some agencies saw the system as being associated with major changes (eg Police) while others simply saw it as the automation of existing processes. In some cases a conscious decision was made to adapt organisational processes to fit the system, as it was less complex and expensive to do this than to adapt the system to fit in with idiosyncratic processes in individual agencies.

- All departments had at least one parallel payrun using the old and new system, but did this in different ways. For example, Tourism, unlike most departments, used the same people to run both systems at the same time. Only one department (Transport) conducted a pilot running before parallel implementation
- Some departments used consultants but others did not. Tourism did not as they wished to keep expertise in-house, while Transport had to due to a lack of staffing.

These differences indicate the substantial impact of the organisational context. Even though the technical system was essentially the same in all agencies, it was implemented in slightly different ways and with slightly different results. It was also important that implementation was not forced on the user organisations, as HRMIS had been, but was planned and achieved by them.

c) Comparison

There were several key differences between the HRMIS and REMUS projects. Perhaps the most significant was that the HRMIS project was generally considered to be a failure while, despite some reporting problems, the REMUS project is broadly considered to be successful. This major difference between the two projects can be attributed to a number of reasons. A significant one is that the relational database technology employed in the latter project was far more mature than that employed in the HRMIS project. The REMUS team were able to choose an already established technical system and only slightly modify it to the needs of the individual departments in the consortium. Other differences which may have contributed to their different outcomes include the level of environmental stability and, most significantly, the way in which the projects were controlled and planned.

The HRMIS project was developed during a period when the parliamentary government changed approximately every two years and several governments introduced large restructuring and redundancy schemes and there was a general movement from a policy of centralised management to a philosophy of "Let the managers manage". In contrast, the REMUS project was developed largely within the lifespan of a single government not noted for implementing major changes. That is, there were fewer major contextual changes which could impact on the project's progress. At the same time, though, those involved in the project actively supported chronological emergence. In the early stages, they only planned early activities and other phases of activity only emerged over time.

HRMIS was initiated and promoted by the two central agencies of Treasury and the Department of Premier and Cabinet, and was largely enforced upon the other agencies. In comparison, REMUS was initiated and promoted by the very

departments who would be implementing it. They had to strongly justify their project to the central agencies who would be providing resources for it and were wary of such projects after HRMIS. That is, the project unfolded in an hierarchically emergent manner.

A centralised corporate information project unit was responsible for the development of HRMIS, while the REMUS project team largely consisted of people from the separate agencies who would implement the system, again reflecting hierarchical emergence. The development of the system was more decentralised, with the separate agencies each having their own project team, with access to the central project team. The aim of this central project team was to coordinate activities between the agencies, provide a central point of contact with the providers of equipment and software and some specialised technical expertise.

Most significant seems to be the background of those directly responsible for developing and implementing the projects. Those who developed HRMIS primarily had an IT background while most involved in the development and implementation of REMUS had a background in human resource management and were mainly seconded from the sections which would be using the system. Many of those interviewed towards the end of the project described it not as an IS or IT project, but as an HR (human resource) project which just happened to use computerised technology and many of those interviewed attributed the success to the project's HR focus. As they were aware of the details of how the system would be used and accepted, they were able to anticipate implementation problems. In other words, they broke down the division between planning and implementation. Certainly, the success of their work was aided by the increased maturity of the technology employed, the stability of the outer organisational context and simply, the experience gained from the HRMIS project. However, one of the most significant lessons from the HRMIS project seems to have been that trying to enforce such a project on others is risky, and this risk was offset by developing REMUS as a consortium, with HR people from the sections involved being directly responsible for the project.

d) HRMIS, REMUS and chronological and hierarchical emergence

REMUS provides some concrete examples of both chronological and hierarchical emergence and can be compared with the HRMIS project, which reflected the top-down nature of much systems development literature. Chronologically, HRMIS was largely planned up-front and its subsequent implementation (or lack thereof) was judged according to these plans. The planning of the REMUS system was far more emergent. The payroll/personnel system began as several agencies combined their resources to investigate potential packages available for their very similar purposes

and gradually grew from there. Only as the new system was being fully implemented did those involved seriously think about the strategic value of REMUS, with the project manager, suggesting this was perhaps a major oversight. Yet could the project have unfolded in the chronologically emergent way it did if the formal, top-down strategic planning model had been followed? The central REMUS project manager commented later he had conducted two projects: one without using the established top-down management approaches; the other using them, and only the former project could be considered successful.

Hierarchically, the HRMIS project also unfolded in a top-down manner, while planning and implementation in REMUS was again emergent, reflecting the closely intertwined nature of hierarchical and chronological emergence. At a departmental level, the choice to join the consortium for REMUS was a voluntary one, governed only by the perceived resource advantages for the agency of doing so. Decisions were made on a consensual basis between the different agencies, with a general recognition that the differing requirements of agencies would have to be catered for. Five years earlier, most of the same agencies had little choice but to be involved in HRMIS, a project initiated by the central budget agencies and developed at a detailed level by a centralised, specialised IT projects unit. Responsibility for REMUS largely remained with the departmental units who would be using it. With the REMUS project, IT people provided some expert advice so the HR people could largely develop their own system and upper management and the executive government provided resources and direction but not directives.

The comparison between the two projects effectively illustrates the difference between chronological and hierarchical emergence and the implications they can have on the process and outcomes. The success of REMUS can be partly attributed to the recognition and active support of chronological and hierarchical emergence while some of the problems associated with HRMIS can be linked but to a lack of recognition of them.

4.3.4 Chronological emergence and the LSP

Hierarchical emergence in the LSP is strongly suggested through the ongoing negotiations which occurred, as discussed in Part 3. Here are descriptions of incidents illustrating how the process unfolded in a chronologically emergent manner, despite the use of top-down models.

a) Delays

Throughout the early part of the project, the CIPU and others emphasised the project had to be completed to a tight schedule to fit in with the OPC's busy workload. For

example, the external project management consultant commented in mid-1995 that the implementation target of early the following year was very tight and allowed for no slippage and the technical systems contractors acknowledged that, "The project has a fixed deadline of March 1996 based on urgent business needs in the Tasmanian Government" (document from technical systems contractors, Summer 1995).

The CIPU broadly kept to their schedule, reporting to the steering committee in mid-December 1995 that implementation was still on schedule for January the following year. There were no formal indications that the project would be delayed and stuck at the detailed design/ testing phase for almost two years. In the middle of March 1995, a delay of two to three weeks was reported. Members of the CIPU expressed concern several times and in April the project leader wrote a formal complaint. Implications of the delays included payment schedules out of kilter with delivery schedules, "unacceptable consequences for the Department of Premier and Cabinet in relation to other projects, the Government's commitments based on the delivery of a successful system by July 1996 and the project's budget".

The Government has made a commitment to the public, the judiciary, and the legal profession for a complete set of Tasmanian Acts to be available in electronic form from July 1996 (as a deliverable from the related Legislation Printing Project). A failure to meet this commitment will create serious embarrassment for the Government (transcripts 17/4/96).

The CIPU came under increasing pressure to quickly complete the project, but had to balance these demands with a need to produce a high quality system which was acceptable to the drafters and others. They received letters of frustration from other departments and organisations, including the Law Society, asking for electronically consolidated versions of acts applicable in their area. The executive manager responsible for the project emphasised the need to speedily complete the project as the constant delays were impacting on support for the project (Project Directors' meeting 20 June 1997).

The continuing delays were documented by an external review of the project:

The project was originally rescheduled from its January 1996 implementation date because:

- the approach of using packages ... for the drafting environment had to be abandoned as the products proved themselves unsuitable;
- use of the alternative ... required the development of an RTF to SGML translator, which proved more complex than originally anticipated.

In late November 1995, [the technical systems contractors] suggested a two phase implementation, to provide functionality for principal legislation by mid-February 1996, and functionality for amendments by early March 1996. This approach was accepted.

Project delivery dates subsequently "crept" a few weeks at a time, with release 1 being delayed a further 10 weeks, and release 2 delayed an (at that time) unspecified amount of time.

In May 1996, an end-date of 31st October 1996 was negotiated for contract purposes.

In June 1996, fixes applied to release 1 were still being retested.

In August 1996, release 2 was scheduled for acceptance testing by October 1996.

At the end of August 1996, it was recognised that the project was not being managed according to a project plan, and so was technically out of control. A full-time CIPU project manager was reinstated.

Delivery dates for release 2 moved week by week until early November 1996. When release 2 was delivered, there were many critical defects. Conditional acceptance of the system is now targeted for April 1997.

This deadline was also not achieved and the project seemed in dire straits. During this time it was difficult for me to obtain access to the project, and much of the descriptions of this time are based on a later analysis of project documentation and interviews with participants.

In mid-1997, following the external review of the project, firm plans were made to implement the system in November. These new plans did not include tight schedules which were likely to slip, but with a renegotiated contract, gave the technical systems contractors time to produce the technically complex system to the required level of quality. This review is discussed further in the next sub-section.

b) Testing and change requests

Several levels of testing were planned for EnAct, but the focus of this section is on acceptance testing. The CIPU project leader emphasised the importance of acceptance testing and commented it was like marriage: "Speak now or forever hold your peace" (transcripts 17/10/96).

Several phases of acceptance testing were held, the first in April 1996, the latest in October 1997. The first session of acceptance testing focused only on the drafting of principal Bills and Statutory Rules and was conducted by approximately half the members of the OPC working in teams under the direction of a consultant. By August 1996, Release 1 of the new system, covering the drafting of principal legislation, was delivered to the OPC. During their acceptance testing of the product, though, they discovered numerous errors. Several drafters later commented they believed much of the acceptance testing they were involved in at this stage was really systems testing. They also expressed frustration that they were not allowed to test the system according to how they would work with it and resented that the systems developers told them what steps to do in detail. They believed drafters were not required to complete this kind of testing and felt resentful that they were not allowed to test the system by using it as they would in practice.

A subsequent in-house review of the first session recommended that the OPC take greater responsibility and be more involved in the testing process as, until then, the CIPU had been controlling the testing procedures. They also commented on the separation between the technical systems and the OPC:

While there have been visits by [the technical systems contractors] personnel to the OPC, and all facilities such as phone and facsimile are available, and a series of weekly project management meeting are held, the project organisation structure limits informal communication and discussion between users and developers (transcripts 18/9/96).

The document outlining the strategy for further acceptance testing sessions was prepared by some members of the OPC plus a consultant and was a different document to the previous version. There was a much greater emphasis on trying to satisfy the OPC that EnAct met "the business requirements as documented in the current version of the LSP Functional Requirements Document". There was also more detail about the scope of acceptance testing activities, the conditions of acceptance, the suspension of testing and the general approach to testing. Acceptance testing was to be completed by two junior drafters and the OPC's executive officer rather than all staff as before. The Acceptance Testing Report later reported it was not appropriate to include the whole office in further acceptance testing due to the OPC's workload and their lack of availability. Thus, unexpected problems and ongoing delays produced many changes in plans and actions, so reflecting chronological emergence.

Acceptance testing performed between October 1996 and January 1997 identified a number of critical system failures. In response to this, the CIPU initiated an independent external review of system quality in mid-1997, as introduced in the previous section. This review pointed out that, while the new system was a "remarkable technological achievement", the technical design for the system and the appropriate documentation were incomplete, system testing plans did not cover important areas of testing and integration, system and acceptance testing was limited by a lack of knowledge, documentation and suitable test data.

The review made a number of recommendations, which were subsequently negotiated between the technical systems contractors and the CIPU as introduced in the previous section. The LSP Project Manager later reported:

The outcome was to reduce the "wheel spinning" by slowing the project down and carefully planning for successful completion. The impact has been significant. Since the revised project schedule was developed in June 1997, [the technical systems contractors] have achieved milestones on time and Tasmania now has a system ready for implementation. (LSP progress report 20/11/1997).

Testing continued informally during this review period and was reconvened formally in August 1997 after further work by the technical systems contractors. Errors were still uncovered, but most were of a minor nature and were corrected. Acceptance testing finished on 24 October 1997 and the system was formally accepted by the OPC on the 19 November.

OPC involvement in acceptance testing was an ongoing issue. Given the delays in the project and the heavy legislative agenda of the government, OPC management felt

they were no longer able to provide the same level of support that had been promised for the planned implementation time. However, the CIPU were aware that such involvement was crucial to the system's success and this was an issue of great debate. As a compromise, only two junior drafters and the office's executive officer were on the acceptance team on a part-time basis for the second session, although others were to be involved for key issues and the final stages. The CIPU believed this was the absolute minimum level of involvement possible, and later this level of involvement was changed renegotiated to two OPC staff members on a full-time basis.

Part of the role of this acceptance team was to keep the rest of the staff informed but several members of the OPC commented later that this did not occur. The two remaining acceptance testers spent a considerable amount of time and effort focusing on the detailed issues arising during acceptance testing and tended to keep their twelve work colleagues informed through ad hoc conversations. They also issued invitations to visit the acceptance testing team as they were testing the system in the CIPU's offices, some two city blocks away from the OPC's offices. At the time, members of the OPC did not feel encouraged to visit the testing site. This was partly due to the high workload of the office, but also due to the management style and culture of the office. Within the OPC, it was important to be seen to be working, and so becoming involved informally in activities outside the physical offices was subtly discouraged. There were also occasions when other members of the OPC were specifically asked for help with particular issues.

The LSP team (which now included the OPC acceptance testing team) did make some effort to keep the members of the OPC informed and involved. On the 29th July 1997, all OPC staff were invited to a demonstration of the new system in action which was given by the two member acceptance testing team from the OPC. They illustrated how it could be utilised by the drafters and the administrative staff and explained the processes the system employed, as partly illustrated in Appendix 3. Very few comments were made in this session and only a few questions were asked. Several members of the OPC did not attend and by the end of the demonstration, only a few remained. This lack of interest may indicate disillusionment with a project which had thus far failed to deliver the promised benefits, though one drafter later commented that this session seemed irrelevant as they had had the chance to see the system in operation on an informal basis. Thus later criticisms that the acceptance testing team had not kept the OPC informed seem unjustified. This also indicates a number of important issues. Firstly, it reflects the ongoing negotiation between the LSP team and the OPC generally. Secondly, it is an indication of the differing perceptions people had of events and circumstances. Thirdly, the need to further

develop the functional requirements reflected the chronologically emergent nature of the process.

c) change requests

Testing highlighted a great number of issues which needed to be resolved before the system was implemented. In all, 422 problem reports and 113 change requests were registered. Issues raised during testing were tracked using an "open issues" log which kept a record of all further actions concerning the issue until it was resolved in some way. Issues were also logged in either a problem or change request log:

An issue becomes a change request if it is desirable and was not included in the requirements specification. Change requests will not affect the acceptance of the system (Acceptance testing strategy v 1 14/3/96: p 5).

The change control procedure involved a formal request for change, an evaluation of the request, including its justification and implication, approval for the change and audit trails. The large number of change requests raised during acceptance testing were attributed to the system's unexpected complexity, overlooked requirements or software defects (External Review 24/2/97).

As far as the CIPU was concerned, many issues were seen to have been negotiated and frozen by this stage and were not open for renegotiation but were to be enforced by OPC management. Acceptance testing highlighted that the signed off functional requirements did not fully meet the needs of the users, but by this time, the document was treated as a baseline document by which to judge further developments of the project. The review document commented,

A concern raised by one member of the Acceptance Testing team was that the Functional Requirements Specification did not properly address the process of drafting. It follows that if [EnAct] does not provide the ideal solution, the software development team would not be entirely at fault. It should be noted, however, that all members of the OPC had input into the preparation of the Functional Requirements Specifications (transcripts 15/10/96).

In essence, the functional requirements document had obtained the status of an *objective fact* whereas before it had been a *subjective issue*, heavily negotiated and discussed. This issue is discussed further in Chapter 4.5.

By the end of 1996, there were strong attempts to limit the number of change requests in an attempt to complete the project.

... in principle we are reluctant to do any redesign at this stage (even though these may be change requests) because even though things may look simple in the context of incorporating them into the rest of the system, they often are not; also more testing is required and any redesign will cause a delay in implementation (letter from technical systems contractors to CIPU 13/12/96).

The external review in early 1997 attributed a third of all change requests to requirements not originally specified or overlooked in the design process (24/2/1997). Thus, even towards the end of the project, the needs of the OPC were being analysed and incorporated into the system, despite considerable efforts to obtain

comprehensive requirements earlier. The review suggested this large number of change requests indicates a "failure in the analysis and design process", yet, as discussed above, it would have been unrealistic to expect all requirements to be uncovered and catered for at the beginning of the project. The problem lies in how we conceive the overall process. As one of the drafters commented on an early draft of this dissertation,

... future projects should recognise that the "functional requirements" should not be a set document. Money should be set aside for changes to functional requirements, because the interpretation of the users is different to the developers. Also, ...users can't give all the information at one time because (1) they have limited knowledge about how developers think and what they require; and (2) developers have only limited knowledge of users. Problems only show up as users are allowed to use the new system (comments on draft 12/3/98).

Even though the Functional Requirements Document was considered to be of a high standard around the time it was produced, it was later viewed as inadequate, especially due to the large number of change requests made. However, it is important to note that it would have been impossible to anticipate all requirements up-front. The requirements had to be identified in an emergent manner, at least to a degree.

d) Implementation

The implementation of the LSP brought to a head the longstanding area concern of user acceptance and LSP utilisation identified by the CIPU throughout the project. Despite considerable efforts to ensure the OPC would be happy with the system, the LSP team were still concerned the OPC would not accept it. The systems developers were in a difficult position during the LSP, being asked by executive management to create a system which the users were wary of and concerns were voiced repeatedly, even towards the end of the project. For example:

The implementation phase of the LSP will be the most difficult to date. The implementation task facing the project team is difficult enough without encountering active resistance within the client group. The next and final phase of the LSP will not be a success unless OPC staff are willing to work together and give the system a chance. The Office as a whole needs to be motivated to provide some degree of united support for the project (Change management plan, November 1997)

In order to pre-empt this risk, the LSP team promoted OPC management's change management plan, the active involvement of the steering committee, tried to ensure the system was thoroughly tested, planned to provide OPC staff with regular updates via email and developed extensive training programs (LSP Business Plan v 0.A 21 November 1997). Upon reading drafts of this dissertation, though several drafters commented that these concerns were over-rated. They stated they knew they had to use the system, given the investment in it and the community benefits from the consolidated database. Some of them remained unconvinced that the system would improve the efficiency or effectiveness of the office's workprocesses, but accepted they had no choice in this matter. This issue of the degree of choice people have during the process of change is pursued further in Chapter 4.5.

Still, at the time of implementation, the management of the OPC continued to emphasise the difficulty of coping with the implementation of the system along with their usual drafting workload. The chief drafter reported to the Steering Committee that:

... the Office of Parliamentary Counsel's capacity to learn and become proficient with the new legislation system while at the same time keeping up with the significant drafting demands placed on them is a significant risk to the project (Minutes of LSP Steering Committee Meeting #9 28/11/1997).

This issue was considered a high risk by the LSP team, both in terms of its seriousness and the likelihood of it occurring.

The LSP team were concerned the drafters would continue working with their current methods and the support staff would enter and format drafts into EnAct, commenting:

Substantial investment has been made in developing a system for drafting legislation. The system is not a purpose built data entry tool which would be more appropriate for administrative staff. Therefore the effectiveness of the EnAct system would be significantly reduced (Project Business Plan v O.A 20/11/97).

The LSP team were unable to predict the likelihood of this risk occurring, but were concerned that some key OPC staff members had been critical of the system throughout the project, a number of drafters had shown little support for the acceptance team during acceptance testing and there was a risk the OPC would react to the process changes required by EnAct. Thus, while at times it had been suggested the drafters would not have to use the system if they did not want to, it was now clarified they would have to.

Due to a growing realisation that members of the OPC were having trouble understanding what the implications of implementing the EnAct system could be, an Impact Analysis Report was prepared by the CIPU and some OPC staff members who had been part of the Acceptance Team, with assistance from others. The aim of the report was to give the OPC an insight into the potential impact of the new system. It did not try and suggest solutions but merely raised issues which needed consideration, as listed in Appendix 2. These covered issues such as ensuring the drafters would use the system, informing OPC clients of delays as the OPC learnt how to use EnAct, the inclusion of externally produced legislation into the system. It helped raise such issues for further consideration, though at the time of writing, most of these issues had not been resolved.

The report also raised issues in relation to training required, ongoing general IT support, handling problems and ongoing issues with EnAct, handling changes to EnAct, system administration and possible future issues. Examples of future issues include ongoing system improvements and changes, measuring the benefits realised by the implementation of the new system, ongoing system operating expenses,

hardware replacement, utilisation of email in the office and the possibility of “paperless” exchange of legislation between the OPC and the Clerks of the House, the increased use of portable computers by drafters, particularly to perform Parliamentary amendments and colour printing of legislation. Further details from this report are included as Appendix 2, but are not analysed here due to space considerations. However, they indicate negotiation was still occurring, issues were still emerging and plans still needed to be developed even while the system was being implemented.

Currently, the plan is that future implementation activities are managed through what the CIPU term an “Outcome Realisation Plan”. This plan aims to facilitate the effective utilisation of the new system and other project outputs, so that the proposed outcomes of the project would be achieved. It is intended to cover areas such as output utilisation (use of the system), management issues, skills and resource requirements, benefits realisation (achieving the proposed benefits of the project, such as improving the drafting throughput), maintenance, “change management” (as discussed above), risk management, consultation and communication and training and staff development (LSP Business Plan v 0.A 21 November 1997). The OPC is responsible for this area of the project and any plan which covers it, although the CIPU are trying to provide support.

In other words, responsibility for implementation may rest with the users, but the way of approaching implementation issues is guided by the developers. This aspect of the project is still unfolding at the time of writing, but it is difficult to see how the users can claim ownership of a plan that is largely created by the systems developers. Although the systems developers would argue it represents only a framework for considering relevant issues, it could be likened to “cultural imperialism” if forced upon the OPC.

The system was formally accepted by the OPC at the end of November 1997 and there was a three month warranty period following this. This period coincided with the summer holiday period, and there were concerns that the system would not be fully used by the end of the warranty period. To date, though, this does not seem to have been a significant consideration. Significant implementation issues have included:

- *the management of process changes.* This issue is still unfolding, but illustrates the fact that systems development does not finish as soon as the technology is “wheeled into” the organisation.
- *the inclusion of all Acts on to the database by the time the system was formally launched.* All Acts had been consolidated manually to 1 February 1997, a date chosen to coincide with the time originally planned for the implementation of the

EnAct system (LSP Progress Report 20 November 1997). This meant that Acts which had changed in 1997 had to be urgently updated and new Acts had to be incorporated so the database would be correct and therefore able to become the “authorised” version of legislation with the commencement of the *Legislation Publication Act 1996*. The OPC expected this would require a substantial amount of work and would be a complex task. At least 62 Amending Acts, 6 Acts containing both original legislation and amendments to other legislation, 11 original Acts and 58 Bills had been addressed by Parliament in 1997 and had to be incorporated into the database.

- *the suitability of the existing desktop technologies.* Due to delays in the system and ongoing EnAct system improvement, the desktop PCs bought early in the project only just coped with the new system and were unacceptably slow. The desktop technology was upgraded in early 1998 to resolve this issue.
- *ongoing system and user support.* This has been negotiated with the department’s IT services branch and further systems improvements are being considered.
- *training and documentation.* Early training covered general computing skills and knowledge, particularly keyboarding and wordprocessing training. Despite this training and the implementation of desktop computing technology within the office some time before the EnAct system was implemented, only a few drafters often used wordprocessing facilities for creating legislation and so required some further general training just before the system was implemented. The first stage of training specifically aimed at EnAct introduced the system and how it could be utilised by the members of the OPC to create principle and amendment legislation. The two members of the OPC who had been responsible for acceptance testing contributed their detailed knowledge of both the new system and existing OPC processes to this training program.

Although the OPC has formally accepted the EnAct system, at this stage it is difficult firmly assess user acceptance of the system. Figure 4.3.2 gives an indication of user opinion of the system just before the system was formally launched in April 1998. One administrative assistant decided to leave the OPC as the system was being implemented, although returned for a period to help the office overcome their short term heavy work load as they completed the database. Three drafters did not find the system acceptable, though one stated he was reserving judgement at this stage. Significantly, two of these drafters had missed large sections of training due to reasons such as illness. These opinions were obtained by asking individuals what they thought of the system, backed up with others’ comments and observations in

February 1998. This figure, as with all other descriptions of the LSP, has also been reviewed by members of the OPC and CIPU.

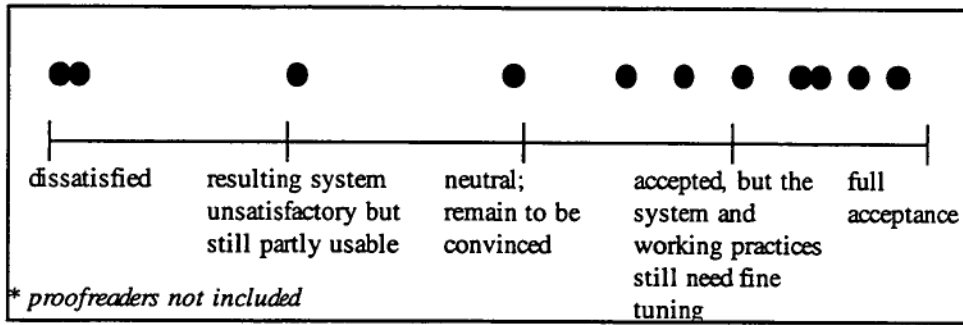


Figure 4.3.2: Opinions of OPC members of the EnAct system soon after implementation

Other comments by members of the OPC included:

- The system was slow and included bugs that needed to be fixed. There were concerns that funds might not be found to do this;
- Some aspects of the system, such as client notes (notes for OPC clients as an adjunct to a draft) were not well developed. This issue was not picked up during testing, possibly due to the differing drafting styles of the individuals involved, or due to the fact that the system was primarily tested by a junior drafter with limited experience with creating complex legislation which might require better client notes. Illustrating the emergent nature of requirements elicitation, the drafters had not previously seen this as an important requirement (see section 2.3);
- Some drafters believed they had not been informed of the known problems with the system, and so were not aware they had to work around them at this stage;
- The system was difficult to use with large and complex documents and most had only used the system properly with smaller documents. There were concerns the system would not be useful for complex drafting;
- The system sometimes restricted what the user reasonably expected to do with it by strictly enforcing structural elements (for eg section numbering, or by restricting individuals' drafting styles);
- Most agreed the system and associated database would prove invaluable for information retrieval in the longer term;
- It was cumbersome to reassign files between drafters and administrative assistants, especially if a drafter and an administrative assistant were working on a document iteratively;
- The drafting staff generally complained about the lack of administrative staff, as the support staff were also working on completing the database before the system was formally launched; and

- Some commented the drafting environment had been overlooked due to a focus on developing database of legislation consolidated on an ongoing basis.

On the whole the responses of the OPC to the system seem more favourable than negative. Several drafters commented on the technical sophistication of the system, with one of them terming it “an electronic version of the Swiss Army pocket-knife” (written comments to chief drafter, 19/2/98). There continued to be ongoing concerns with the system, but all staff members were using it, at least to a degree.

The EnAct system was publicly launched on the 9th April 1998, some two years after the initially planned time. The launch had depended on the implementation of the EnAct system, the completion of a database of current legislation authorised by the CPC, the commencement of the *Legislation Publication Act 1996* and the implementation of a web site to allow public access to legislation, created as part of the associated Legislation Publication Project (see www.thelaw.tas.gov). This formal launch of the system ended my research involvement with the LSP. The third phase of the LSP, focusing on implementation, continues, but details of it are not crucial to the primary research themes of this dissertation.

4.3.5 Reflections: Emergence in theory and practice

Upon reflection, the EnAct system resulting from the LSP will probably be considered a success. However, the project which developed it cannot be considered totally successful as the system was delayed for two years and the project came very close to being abandoned. The CIPU remain concerned that the OPC will not accept the system because basically it was forced upon them and the OPC perceive that they gain very few benefits out of the system themselves. The systems developers acted in good faith and attempted to adopt recognised standards of best practice in both systems analysis and project management, so the question must be asked that perhaps there is something wrong with these standards.

The plans for the Legislation Systems Project were aligned with most of the normative literature in the area and reflected its top-down assumptions. The project was initiated and strongly supported by executive management, just as the literature suggests. The systems developers aimed to align their practices with recognised standards of best practice, by buying a comprehensive methodology, employing consultants, and achieving ISO certification. The people involved did all they reasonably could to ensure that their practices were aligned with recognised standards of best practice. As an external review commented,

While the management of the project has been carried out by capable individuals using sound methods, this project has suffered from a number of deficiencies (transcripts 24/2/97).

Being primarily human resource management people, those involved in the REMUS project did rely so much on the systems development literature but, perhaps in this case, it was beneficial that they did not. While the systems development literature has strongly promoted the use of comprehensive methodologies, a similar systems development project in the British Columbia, Canada, cited one of the reasons for their success as being a decision to not adopt existing methodologies based on the waterfall or lifecycle model. They commented:

Most government systems projects are executed using a proprietary methodology... Typically these methodologies are based on a life-cycle ("waterfall") approach or an information engineering approach. They generate volumes of diagrams and CASE output, giving the impression that the authors are paid by the amount of paper produced. They require that the users pore over data models and functional models, and that the people somehow relate these to their jobs.

In the BC Government two large failed projects... went on too long, were over budget, and the many organizations involved had difficulty achieving consensus on detailed business requirements. Eventually the projects were cancelled. They produced a lot of paper, though.

Why does this happen over and over again in large government projects? Could it be that the traditional and proprietary methodologies are not the path to success? (CHIPS 1996: p 40)

On commencing observations of the LSP, the goal was to track a project from its planning stages, through its development to its implementation and subsequent use. The aim was to see how the process of embedding an information system in its organisational context actually occurred and compare it against the normative literature in this area.

With the ongoing delays, what I have been able to observe is that, with the best intentions and effort, the dominant models for planning, developing and implementing an information system have not produced the intended results. While these delays have prevented observations of the whole process within the time allowed for this research project, these delays are indicative of the issues being raised in this dissertation. The LSP team put a great deal of effort into aligning their actions with recognised standards of best practice, yet perhaps these dominant standards of best practice are not appropriate.

As Fitzgerald (1995) noted, the strict use of methodologies can lead to goal displacement, where developers become preoccupied with developing a methodology at the expense of actual development. DeGrace and Stahl (1990 in Fitzgerald 1995) analysed the documentation of several development projects and noted that 90% of it was devoted to reporting the status of the project, while less than 10% described what was to be done and why. As Fitzgerald suggested:

Systems development in practice is actually an unstructured, evolutionary process yet development methodologies attempt to impose complete solutions, when the real nature of systems development is not well-understood (p 15).

This description of the LSP over three and a half years has analysed this "real nature" of systems development and has found it to be chronologically and hierarchically an

emergent process involving substantial negotiation. This stands in contrast to the bulk of the systems development literature, which primarily views the process as a top-down one, and ignores the socially negotiated nature of such changes.

On a surface level, continuing delays to the delivery of the EnAct system can largely be attributed to unanticipated technical complexity. This is all very well in hindsight, but how can this observation aid similar situations when people are making decisions without a crystal ball?

One possibility is to recognise that the implementation of novel technology is risky, so it should be avoided. Harrington (1991) adopts this view, stating,

Solving efficiency problems with information technology is like putting a lion in the sheep pen to keep the wolves away. It can work, and sometimes successfully, but the knack is to ensure that the lion does not eat the sheep in the meantime (p 237-8).

However, this is unrealistic. Firstly, innovative information technology provides too many opportunities to let them slide by. Secondly, even if the technology has been employed in other situations, and the technology employed in the LSP largely had been, each application of it in a differing organisation context is innovation.

Computerised information systems development generally involves at least some novelty, even if the technology deployed is well established. This is because the processes surrounding the technology also need to be considered and the organisational context of information systems development is never exactly the same, a point underlined by the differing implementations of the same technology in the REMUS case study.

During the LSP, those involved tried to mitigate these risks by techniques and concepts such as risk analysis, change control procedures, concepts of change management and so forth, as advised by the best literature and consultants they could find. These are generally standard practices, but are they enough?

I suggest they are merely “cover up” solutions to what are essentially fundamental problems in how we define information systems development projects and conceive the process of developing and implementing them. Observations that ‘major project delays occurred in the LSP due to unexpected technical complexity’ identifies a problem, but it does not suggest a solution. A solution lies not with the technology (use more established technology) but in the way we conceive the process of systems development. The process must be viewed as chronologically and hierarchically emergent if we are to recognise ongoing social processes at a micro level, the closely intertwined nature of planning, development and implementation and also the manner in which change unfolds at an institutional level, as discussed in Chapter 4.5. This emergence is fundamental in the multi-level process model described here and should inform the development of more effective macro models of the process.

Up until now, the aim of this dissertation has been to introduce, explain and illustrate this point. Yet it is all very well to critique existing practices. Most systems developers would adopt these dominant models because they know of no other. What they need are alternative process models of information systems development which do not suffer from the problems of the dominant SDCL and this is pursued in the next chapter.

Hierarchical and chronological emergence avoid the three primary problems identified with the SDLC model, as illustrated in Figure 4.3.3. In a practical sense, they recognise the planning and management of change in organisations, but do not overemphasise them at the expense of bottom-up or backward-mapping influences on change. Hierarchical emergence recognises the context of development is crucial and that user participation or ownership is an appropriate way to ensure contextual issues are considered. That is, a recognition of social processes allows a better consideration of contextual issues. Effective user involvement requires negotiation of meaning and interests if the social processes are to unfold in a way that allows relevant contextual issues to be addressed, thus these social processes strongly reinforce chronological emergence. Chronological emergence also involves a recognition that the context is not only important, but that it is dynamic and so plans for action cannot be precisely made up-front. Thus any alternative models to the systems development lifecycle model should reflect both chronological and hierarchical emergence.

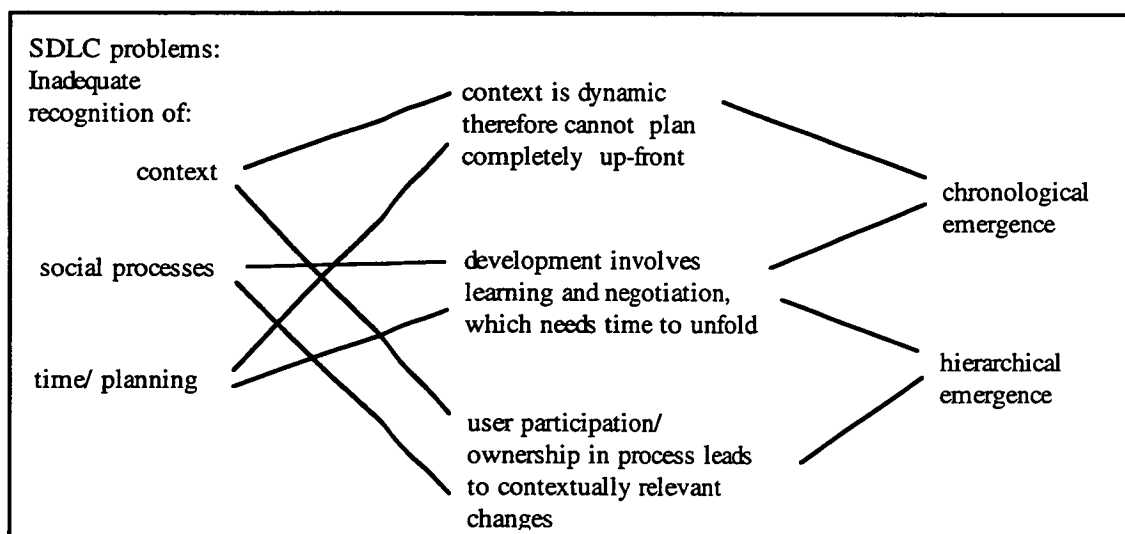


Figure 4.3.3: The relationship between problems with the SDLC model and chronological and hierarchical emergence

This has been a complex chapter, but a crucial one. As well as bringing to an end descriptions and analysis of the LSP, it also signals completion of critiques of systems development practice. In the next chapters, alternative normative process models operating at the *dasien* level and a process model at the level of *longue durée* are suggested. In summary, this chapter has made the following key points:

- The relationship between planning and implementation is a key one when considering the process of change. Three broad approaches can be identified: top-down, bottom-up and emergent. All three are discussed at length in the policy analysis literature, but only the emergent approach seems to have descriptive validity. This is also reflected in systems development and implementation, despite the dominance of top-down approaches.
- An examination of literature focusing on this relationship between planning and implementation identifies some confusion as to what is precisely meant by it. It could refer to the hierarchy of the organisation (ie the relationship between those who plan changes and those who implement it) or to chronology (ie, how plans are implemented over time). I argue it is useful to label these separately and have referred to them here as *chronological* and *hierarchical* relationships.
- Chronological and hierarchical emergence was illustrated through a description and analysis of the HRMIS and REMUS case studies and the final stages of the LSP. Part 3 had illustrated the process of systems planning and development emerged hierarchically through negotiation and associated social processes. Here it was illustrated that, despite efforts to make the LSP unfold in a chronologically top-down manner, the process was chronologically emergent. The project was substantially delayed, testing revealed numerous change requests and implementation activities still involved a great deal of planning.

Aligned with most normative literature and practice, those involved with the LSP primarily employed a top-down conception of the process. This process did not adequately reflect their actions and can be linked to problems encountered, such as the ongoing delays and perceptions of user dissatisfaction. The system resulting from the LSP will probably be considered a success but the process by which it was achieved cannot be considered so. It could be argued that the ends justify the means and, as the ends were achieved, the means can be considered successful. However, there are alternative conceptions of the process which could be employed, as the next chapter illustrates. Chronological and hierarchical emergence lays the foundation for their consideration.

4.4 Alternative macro models of process

There is little point in only criticising the dominant SDLC model if alternative approaches are not suggested. This chapter evaluates some existing alternative models in terms of their ability to reflect chronological and hierarchical emergence and so be compatible with the micro and institutional levels of process described here. In practice, chronological and hierarchical emergence entails iteration, incrementalism and participation, and these existing models are evaluated in terms of these criteria. Several possibly viable alternative models are suggested, including Rapid Application Development and STEPS (an elaboration of the ETHICS approach). Alternatively, the process of systems development could usefully be conceived as a series of interlinked projects, rather than a monolithic one.

Why, if there are so many problems with the SDLC model, is it still being used? It could be argued that the model is not a descriptive one but a management tool for rationalising people's actions (Buchanan and Boddy 1992) or that the model has become so entrenched it is simply accepted as an objective fact (Lewis 1994). Arguments such as these are not disputed here, but there is a more basic and fundamental reason: alternative models are rarely systematically and critically compared against the SDLC model. In short, systems developers are often unaware of alternatives. For example, Nandakumar (1993) usefully highlighted the inadequacies of the SDLC model and introduced possible general models to describe what actually happens (such as structuration theory), but did not investigate if there were any existing alternative models which addressed the issues raised by his analysis. Others suggest an alternative model, yet it is difficult to find evidence that such models have been critically evaluated by other authors or that they have had the chance to be widely diffused (eg Eason 1988). Examinations of alternative models in texts in the area tend to be unsystematic or strongly biased towards the SDLC model or the authors' own creation. Here some alternative macro models of process are evaluated.

4.4.1 Chronological and hierarchical emergence in practice

As discussed in the previous chapter, chronological and hierarchical emergence would help avoid the problems associated with the SDLC, but what does this mean in practice? A process model which reflects both chronological and hierarchical emergence will exhibit three primary attributes:

- iterative development (prototyping);
- incremental development; and

- participation or user-led design.

Iterative and incremental development supports chronological emergence while iterative development and participation or user-led design aids hierarchical emergence by strongly supporting the negotiation of interest and meaning which underlies systems development. An analysis of 24 common texts on systems development reveals that only four of them reflect all three attributes of hierarchical and chronological emergence, though many of them support one or two. A summary of this analysis is provided as Appendix 1, and is discussed further below.

a) Iterative development (prototyping)

Prototyping, or iterative development, supports chronological and hierarchical emergence by providing users and others with a view of the system before it is implemented so it can be discussed and reviewed. There are two ways of using prototyping: either as a technique within another methodology, or as the basis for an alternative model to the SDLC (Krogstie 1995).

As a technique, prototyping can be used as a one-off, throwaway tool for developing requirements, or as an alternative methodology to the SDLC approach. Prototyping can be used as a tool to aid the determination of requirements within the SDLC model, as it was with the LSP. This is one way of bypassing some of the user/system developer communications problems attributed to the dominant conventional model.

Prototyping is also often presented as an alternative to the SDLC model (eg Eliason 1990; Laudon and Laudon 1995; Martin 1995). As a methodology, it is highly iterative and is characterised by the ongoing creation and use of prototypes so that feedback from users and others can be incorporated into the system early in the process (Krogstie 1995). The prototype is revised until it is acceptable and then is implemented. In this way, the functional requirements of the system can be validated and communication between all parties is improved. Changing user requirements are accepted as a norm and are handled through interactive generation and validation of functional prototypes. Vonk (1990 in Krogstie 1995) suggests that the primary benefit is that it reduces uncertainty, and so it is suitable for projects in which there is a great deal of it. Carey (1990 in Krogstie 1995) suggests that advantages of prototyping methodologies include faster development time, the creation of easier to use interfaces, less human resources required to develop systems, decreased backlogs and improved communication between users and developers. Disadvantages include the fostering of unrealistic user expectations, as what the user sees may not be what the user gets.

In practice, it is difficult to differentiate prototyping as a specific methodology because it can be such an integral part of other methodologies, such as RAD. Just over half of the 24 common texts surveyed discussed prototyping or iteration in a significant manner (see Appendix 1).

b) Incremental development

Incremental development constructs a partial system, implements it and then progressively improves functionality or performance (Krogstie 1995). The advantages of this approach is that costs are kept down before an initial system is ready to use, the time required to produce a system is shortened, and, because of this, the possibility that user requirements will change is reduced. Krogstie (1995) claims that incremental development presumes that most of the requirements are understood up-front and that the choice is made to implement only a part at a time. Thus incremental development can be distinguished from prototyping, as the former reflects emergent implementation, but not necessarily planning, while prototyping is planned and designed in an emergent fashion, but may be implemented in a one-off fashion. Avison and Fitzgerald (1995) termed this approach “evolutionary development” and noted it generally assumes that systems maintenance is not a cause of inadequate processes of development, when corrections are made, but an integral part of the ongoing process.

Like prototyping, it is possible to use incremental development within a range of process models. Rapid Application Development not only promotes prototyping, but also incremental development, for example. The LSP team adopted incremental development broadly within the SDLC model, as suggested by Thompsett (1993). With this strategy, the system is broken into semi-independent subsystems. These subsystems can be sequentially released, with different and progressively more complete versions of the system are implemented, or they could be released in a concurrent or parallel fashion. This strategy is chronologically emergent, but only in respect to implementation. While the system may be implemented incrementally, planning occurs monolithically, and only occurs incrementally insofar as plans are articulated in more detail.

Texts surveyed were deemed to support incremental development if they reflected at least two of the following attributes:

- changing requirements and managed for it;
- planning is an ongoing process;
- no strict separation between planning and implementation; and/or

- modular development and/or incremental development or incremental strategies were encouraged.

A third of the 24 common texts surveyed did so, suggesting incremental development is less widely disseminated than iterative approaches. It is important to note, though, that chronological emergence entails incrementalism as well as iteration.

c) Participation or user led design

If we recognise that users play an active role in systems development, then their participation becomes an important facet of the process and the highly negotiated nature of the process is underlined. These issues have been addressed in Chapter 3.2. Texts were deemed to support active participation if they supported two or more of the following issues:

- users have active involvement or influence throughout the process;
- the social processes surrounding systems development are acknowledged; and
- systems developers were seen as change facilitators rather than experts.

Just under half of the surveyed texts supported active participation.

4.4.2 Alternative models of process from the systems development literature

There are a number of ways of identifying alternative process models. One approach is to examine what processes are used in real-life projects and construct a typology of them. This approach was adopted by Sabherwal and Robey (1995), who identified six distinct archetypical approaches in 53 separate projects. These included:

- the textbook lifecycle approach;
- a logical minimalist approach, which was very like the first one, but with little project definition;
- off-the-shelf purchasing;
- outsourced co-operative, where the system was developed by internal and external developers;
- problem driven minimalist, where there was little project definition or reassignment of organisational roles; and
- in-house trial and error, which starts as a lifecycle approach but results in frequent modifications in response to problems.

While usefully identifying that the SDLC does not cover a large range of projects, the creation of such a typology raises more questions than it answers. As with many

quantitative studies, it ignores some pertinent contextual issues. Did the sequence of events used to develop the typology unfold because of the way the systems developers conceived the process or visa versa? What conception did people involved in the projects have of the process? How did this differ from Sabherwal and Robey's analysis? Why did it differ?

An alternative way of identifying other possible process models is through a review of recent literature on systems development. Recent general texts on systems development²⁵ identify a number of alternatives to the SDLC approach. While approximately half of those texts suggest no alternative to the SDLC approach, many others suggested prototyping as an alternative to the SDLC as well as a tool to be used within the SDLC model (eg Eliason 1990; Avison and Fitzgerald 1995; Martin 1995; Hoffer, George et al. 1996). This involves the iterative development of a system through the use of one or a series of prototypes and a range of prototyping methodologies are identified. Other alternatives outlined in these and other texts include:

- object oriented approaches to development (eg Taylor 1992; Dewitz 1996; Hoffer, George et al. 1996);
- the use of off-the-shelf packages (Laudon and Laudon 1995);
- soft systems approaches, particularly Soft Systems Methodology (SSM) (Checkland 1988; Lewis 1994);
- joint application design / rapid application development (JAD/ RAD) (eg Avison and Fitzgerald 1995; Dewitz 1996);
- development by end users (Laudon and Laudon 1995);
- participatory approaches (eg Mumford 1995);
- socio-technical systems approaches (eg Eason 1988; Avison and Wood-Harper 1986);
- systems "devtenance" (Krogstie 1996);
- the opportunistic model (Khushalani et al 1994);
- stroke-wise development (van Slooten and Schoonhoven 1994);
- the "V" process model (Smith 1997); and
- the "adaptive" lifecycle model (Highsmith 1997).

²⁵ ie published since 1990; see Appendix 1.

Several of these alternatives cannot be separately identified while several contain very little concept of process over time, so cannot be considered alternatives to the SDLC. Using off-the-shelf packages, for example, does not include any conception of process over time in itself and is generally included as an option from within the SDLC model instead of detailed design. SSM only really covers problem definition activities and so cannot be viewed as a model of the whole process. Again, Khushalani's opportunistic model only considers the process of design and shall also not be considered here. Van Slooten and Schoonhoven's (1994) strokewise model of development is not unlike Boehm's spiral model and will not be considered separately.

Thus, nine possibly viable alternative separate models are identified:

- Boehm's spiral model;
- Object oriented approaches;
- Krogstie's systems devtenance;
- Rapid application development (RAD);
- Participative approaches, such as Mumford's ETHICS;
- End user computing;
- Multilinear models, such as Eason's sociotechnical design;
- Avison and Wood-Harper's Multiview (Avison and Wood-Harper 1985, Avison and Fitzgerald 1995);
- the "V" process model (Smith 1997);
- The adaptive lifecycle (Highsmith 1997).

This is not intended to be an exhaustive examination of possible alternatives, but an investigation to see if there are possible process models in the systems development literature which overcome the problems with the SDLC model. These possible alternatives are described in more detail in the following sections and evaluated, illustrating that Krogstie's systems devtenence, Rapid Application Development and an extension of Mumford's ETHICS methodology reflect both chronological and hierarchical emergence.

a) Boehm's spiral model

Boehm (1988) developed a framework for systems development where risk analysis determines the choice of more specific methodologies. In other words, Boehm's model subsumes most other process models. As software is being developed, the idea is the project progresses through a cycle of determining objectives, alternatives and

constraints, evaluating alternatives and resolving risks, developing the product and planning the next phase. The utility of prototyping is acknowledged, the model supports both incremental and iterative development. However, user involvement is barely mentioned, suggesting that hierarchical emergence is not supported.

b) Object oriented approaches

Object oriented approaches exhibit the following characteristics:

- the system may be analysed and specified at each stage of the development process;
- reuse of existing components is encouraged; and
- system maintainability is enhanced through the principles of information hiding and inheritance (Krogstie 1995).

Krogstie comments that object oriented approaches support bottom-up development and perhaps also a combination of bottom-up and top-down development, but it is unclear if he is referring to chronological or hierarchical emergence or both.

Slonin (1994 in Krogstie 1995) reported that there were more than 150 object oriented approaches, with separate theory, terminology and modelling approaches and no clear leader or dominant approach. It is difficult to differentiate object oriented approaches from other ones, primarily because different authors in the area combine object oriented approaches with different models of process. For example, Montgomery (1994) uses it in conjunction with James Martin's Information Engineering methodology (broadly a SDLC approach), Taylor (1992) combines it with prototyping while Dewitz (1996) combines it with Joint Application Design (JAD). Hence, the object oriented approach does not seem to be aligned with a different model of process in itself.

c) Systems devtenance

Surveying Norwegian firms concerning their IT projects, Krogstie (1995) noted that organisations tended to spend about 60% of their IT costs on maintenance. He suggested the whole process of developing and maintaining an information system should be considered as one process, which he termed systems devtenance. Although only briefly articulated and not widely diffused, this model provides an alternative to the SDLC.

Systems devtenance is "a more comprehensive lifecycle that includes maintenance, uniting software development and maintenance functions by sharing the same tools and methods in both" (p 13). He suggests that it is difficult to differentiate between ongoing maintenance and "new" application systems which are replacements and, while the process involves four distinct phases centred around the use of conceptual

modelling, the process is not viewed as a sequential one. Krogstie also promotes extensive and active user involvement and is based on constructivist (subjectivist) assumptions. In other words, Krogstie's model of systems devtenance supports iterative, incremental and participative development and so, although only briefly articulated, does support chronological and hierarchical emergence.

d) Rapid Application Development (RAD)

As its name suggests, Rapid Application Development (RAD) involves the speedy development of systems in order to cope with rapidly changing business contexts. This version is strongly associated with the Information Engineering methodology but others use it separately. RAD is basically a combination of tools and techniques based on an evolutionary prototyping approach. It aims to identify people who could influence or be influenced by the system or project and involve them in the process via workshops in the early stages of development. RAD does not just support iteration through the extensive use of prototypes, but also incremental, or evolutionary development. One-shot systems development projects might include various stages of implementation which are planned incrementally through cycles of development.

There are a variety of RAD methodologies, but perhaps the most well-known is James Martin's version (Avison and Fitzgerald 1995; Hoffer, George et al. 1996). In James Martins' version, there are four phases to the RAD process (Avison and Fitzgerald 1995). The first, requirements planning, is to involve all key people so their commitment is obtained to the project. Secondly, users basically design the system with the aid of extensive prototyping to aid communication. A workable basic system, or the core elements of it, is then to be quickly constructed in a four-to-six week period, after which the system is implemented in the user organisation. The system is then refined with further models developed and integrated progressively. Hence, the process is a cyclical, evolutionary one, with each cycle aiming to take about 90 days. Each iteration focuses on the most critical outcomes.

An important part of RAD is Joint Application Design (JAD). JAD was developed by IBM in the late seventies in an attempt to bring more structure into the development of systems requirements. Basically, it is a series of structured meetings for users, managers and developers. The primary idea of JAD is that all those who can influence or who are influenced by the project should be in the same room at the same time to discuss systems requirements, design details, resources and build up shared understandings. It is important to ensure the right people involved and there should be an executive sponsor who can help over-ride bureaucratic issues and issues of power which could hinder rapid development. JAD sessions can suffer from the usual problems involving group dynamics, such as issues of power and access to the floor,

but attempts are made to offset these through appropriate facilitation and communications technologies (Avison and Fitzgerald 1995; Hoffer et al. 1996).

As with prototyping generally, when using RAD, one can overlook software engineering principles and control issues. In the rush, there can be inconsistencies between modules, a lack of component reuseability, documentation and testing and non-compliance with standards (Hoffer, George et al. 1996). Laudon and Laudon (1995) suggest that while prototyping is appropriate when there are unclear requirements, it is difficult to apply to large, complex systems, unless it is separated into components, especially if there are large numbers of users or a great amount of data.

One advantage of RAD is that the context is carefully considered via the active involvement of all those who will be affected or who can affect the development process. However, it ignores one aspect of context crucial in the LSP – users would not have been happy with a rough solution. Through its strong emphasis on social interactions at crucial stages in the project, social processes and hierarchical emergence is also encouraged to a degree (depending, of course, on how the process is facilitated). It also supports chronological emergence in that the split between planning and implementation is reduced.

Thus, RAD supports incremental, iterative and participative development. RAD and prototyping allows a greater consideration of the context through an iterative approach. The context is allowed to change and contextual issues are raised through the involvement of key people. It includes all key players in crucial parts of the process and emphasises the role of a facilitator to ensure group decision making processes work effectively. By bringing everyone involved into the same room at the same time, the social processes surrounding systems development are dealt with directly. Development cycles are kept short and reiterations of the cycle are allowed. By viewing development as a series of cycles dealing with the most crucial issues each time, rather than as a single learning cycle, RAD caters for chronological emergence. Extensive use of prototyping breaks down a division between planning and implementation activities. Also being well articulated, it provides a viable alternative to the SDLC approach.

e) Participatory approaches- ETHICS and STEPS

Participatory design approaches have been identified as "a viable alternative to the SDLC" (Hoffer et al. 1996). One of the most well-known participative approaches is Mumford's Effective Technical and Human Implementation of Computer-based Systems (ETHICS) (Mumford 1995). Mumford comments that models of the computer systems development process became structured and formalised very

quickly and accepted uncritically by systems developers. ETHIC is based on socio-technical design approaches, where there is a strong recognition that technical systems cannot be developed in isolation from their social surroundings. ETHICS has three main objectives:

- to enable future users of a system to have an active role in the design of the system and the work structures surrounding it;
- to ensure the system is acceptable to users by increasing both their effectiveness and job satisfaction; and
- to assist users to become increasingly competent in the management of their own organisational change so that this becomes a shared activity with the technical specialists and reduces the demand for scarce resources (Mumford 1995: p 27)

Participation by those who will be affected by the system is a core element of this approach, and there is a strong emphasis on the need to negotiate change. Users are considered particularly important in the process because they have so much say in determining the project's success and are probably the most knowledgeable about existing practices & procedures. Mumford comments:

Today the reality of most change situations is one of negotiation. There is a recognition of different interests... successful change requires the identification and resolution of conflicts of interest (p 25).

Thus Mumford explicitly sees development as a process of negotiation of interest and learning. Hence the social processes surrounding systems development and implementation and hierarchical emergence are strongly recognised. ETHICS' strong focus on participation and user-led design, along with a consideration of the role of a facilitator, makes the consideration of social processes a strength of this approach.

However, chronological emergence is not explicitly reflected. The methodology is a one-off development cycle with a set of "systematic steps" outlined:

- Diagnosing user needs and problems, focusing on short- and long- term efficiency, job satisfaction and quality.
- Setting efficiency, effectiveness, job satisfaction and quality objectives.
- Developing a number of alternative design strategies which will assist the chosen efficiency, effectiveness, job satisfaction and quality objectives.
- Choosing the strategy which best achieves all of these objectives.
- Choosing hardware and software, and designing the system in detail.
- Implementing the new system.
- Evaluating its success once it is operational (Ibid: pp 28-29)

Such systematic stages help to maintain control of the process and communication between those involved, but it does suggest a linear conception of the process, with a strictly sequential break between planning and implementation. The methodology reflects a largely sequential conception of process, with an implicit division between

planning and implementation. This gap is reduced by the strong user involvement in the planning and design stages. As Crowe, Deeby et al. (1996) comment, participative approaches offset conflict situations by avoiding the problem of hand/brain separation implicit in most process models. However, while the context of the system is considered through the extensive involvement by users, changing contextual circumstances are not catered for. Thus, ETHICS heavily supports hierarchical, but not chronological emergence explicitly.

Mumford's approach has been further developed by researchers at the University of Berlin, who have created STEPS, or Software Technology for Evolutionary Participative System Development (Floyd et al 1989). While Mumford's ETHICS approach supports participation, it does not support incremental or iterative development. STEPS provides support for prototyping and cycles of development, so supporting iteration and incremental development. This elaborations of ETHICS provides a possible alternative to the SDLC model as it supports both chronological and hierarchical emergence.

f) End user computing

End user computing has been identified as another alternative to the SDLC (Eason 1988; Laudon and Laudon 1995), though , less than half of the current systems development texts surveyed mentioned end user computing or user development as an alternative (see Appendix 1). The actual process by which users develop their own computer systems is rarely investigated, reflecting perhaps its largely *ad hoc* nature. Alter (1996) suggests a number of broad phases of end user computing based on the assumption that the end user develops their own system and are responsible for the results. Since the user will develop the system, formal specifications are rarely developed. The users develop the system themselves, using tools that do not require a professional level of knowledge. Implementation is simplified as the developer is the user and operation, maintenance and ongoing development is their responsibility. End user developed systems are generally very small in scale and tend not to be complex, multi-user or mission critical ones. The reason for this is that end user systems often do not comply with technical standards or perhaps organisational ones.

The process of end user computing development can thus be described as hierarchical and chronologically bottom-up, rather than incremental, and suffers from many of the same problems identified with the bottom-up approaches to policy implementation (See Chapter 4.3).

g) Eason's socio-technical systems development

Believing that we are unlikely to achieve real benefits from technology if human and organisational issues are not catered for and that existing system design processes do not adequately deal with them, Eason (1988) and others at Loughborough University's HUSAT research centre have developed an alternative process model. As well as criticising the SDLC approach, they were also critical of Mumford's ETHICS and other participative approaches and end user computing. They were critical of approaches such as ETHICS because the process depends a great deal on an expert to guide users through a very complex process and because the ETHICS approach is very different from previous systems development methods and so was difficult to adopt.

Eason's alternative process model is a multi-linear one based on socio-technical principles. He sets out ten propositions the model is intended to address:

- The successful exploitation of information technology depends upon the ability and willingness of the employees of an organisation to use the appropriate technology to engage in worthwhile tasks.
- The design target must be to create a socio-technical system capable of serving organisational goals, not to create a technical system capable of delivering a technical service.
- The effective exploitation of socio-technical system depends upon the adoption of a planned process of change that meets the needs of people who are coping with major changes in their working lives.
- The design of effective socio-technical systems will depend upon the participation of all relevant 'stakeholders' in the design process.
- Major benefits will only result if the socio-technical developments are directed at major organisational purposes where there are opportunities to be taken or problems to be resolved.
- The specification for a new socio-technical system must include the definition of a social system which enable people in work roles to co-operate effectively in seeking organisational purposes and provides jobs which incumbents perceive as worthwhile.
- Information technology systems must be designed to serve the functional needs of the organisation by serving the functional needs of individual users in a useable and acceptable way.
- The effective exploitation of information technology requires a major form of organisational and individual learning.
- The exploitation of the capabilities of information technology can only be achieved by a progressive, planned form of evolutionary growth.
- To be successful, socio-technical design concepts must as far as possible complement existing design procedures and organisational change practices (pp 44-49).

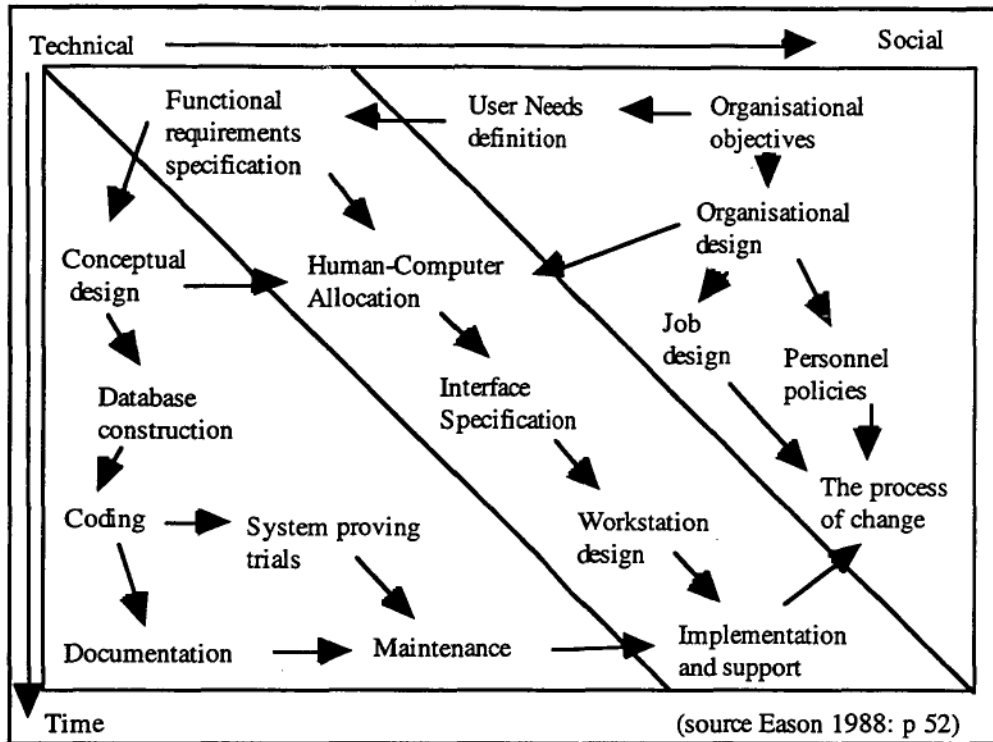


Figure 4.4.1: Eason's design topics in the socio-technical design of IT systems

In response to these propositions, Eason suggests a "toolbox" approach to systems development which is not intended to be part of a "one-shot" implementation process. Figure 4.4.1 illustrates the multi-linear form of this model. There is a broad sequential flow to the model, although some tasks are completed in parallel. Eason suggests that the social aspects of systems design should be the province of the users while technical systems developers are concerned with the technical aspects. Both groups need to be concerned with tasks located in the central part of the model where social and technical concerns interact.

Eason acknowledges that this model still shows signs of "one shot" implementation efforts and suggests that evolution and iteration can be built onto this model in two ways. The first involves running through the large-scale design process multiple times through the development of prototypes, pilot systems and gradual implementation strategies. The second is nesting the process in wider processes of IT policy development and to allow end user customisation within the process, so that the same conceptual process is used at a number of levels. Thus, chronological emergence is not reflected in this model, though Eason suggests that incremental and iterative approaches can be added.

Eason spends a considerable amount of time focusing on the relevance of the relationship between different parties involved in the process of systems development and their relationship to the process itself. Hierarchical emergence is included but is

limited due to the largely chronologically top-down nature of the process, as it assumes users' needs can be fully defined up-front.

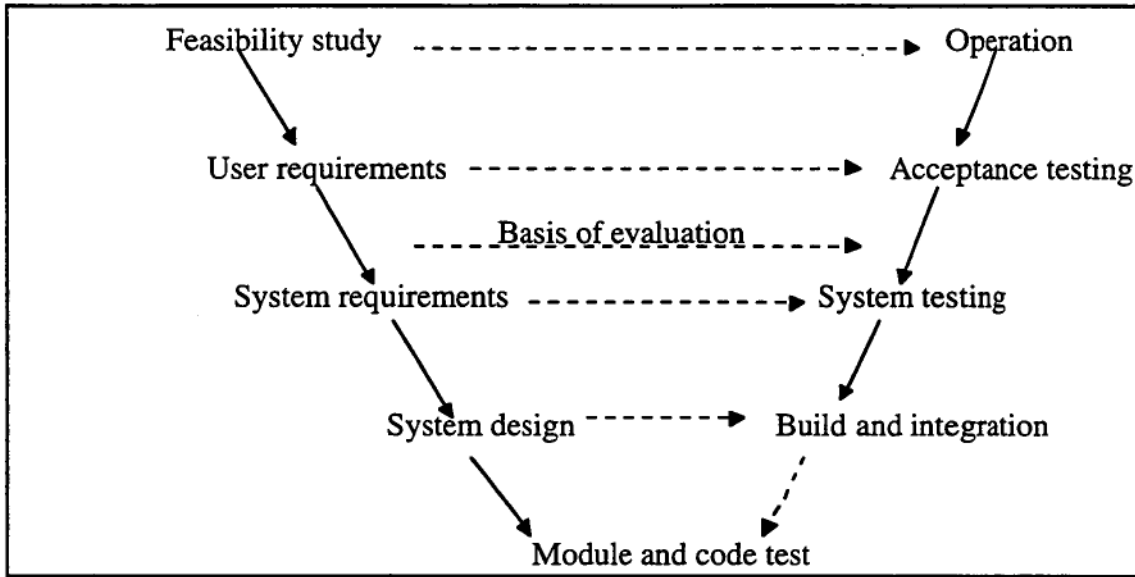


Figure 4.4.2: the V process model (Smith 1997: p 135)

h) Multiview

Like Mumford's ETHICS, Avison and Wood-Harper's (1985, Avison and Fitzgerald 1995) Multiview methodology is a socio-technical approach to systems development, and so is very similar in many respects. As with ETHICS, Multiview fosters an initial focus on the human aspects of systems development before considering technical issues. In simple terms, the five stages of Multiview are:

- Analysis of human activities;
- Information modelling;
- Analysis and design of socio-technical aspects;
- Design of the human-computer interface; and
- Design of technical aspects.

However, there are several significant differences between the two approaches. Firstly, Multiview places less emphasis on participatory issues and tries to incorporate significant elements from some of the "harder" systems development methodologies, such as STRADIS and Information Engineering. Secondly, it recognises that there can be no one approach to systems development which is suitable for all types of situation. Multiview is, essentially, a contingent approach, which incorporates different approaches for different situations. However, as with Mumford's ETHICS approach, the methodology finishes with the design of the technical elements of the system, and does not consider implementation. Thus chronological emergence is not

reflected in this approach, and, while hierarchical emergence is suggested, it is not emphasised to the same degree as Mumford.

i) The “V” model

The V process model, illustrated in Figure 4.4.2, aims to illustrate how phases of the lifecycle can operate in parallel (Smith 1997). Information can flow to several stages at once. There is testing at each stage to reduce the need for iteration and users are to be included in the specification and building phases. Smith notes, however, that it is difficult to go back to rectify errors and much time is spent investigating each phase. Thus the V model does not seem to support chronological emergence. It also suffers from a lack of detail, with no further references provided.

j) The adaptive lifecycle

Highsmith's (1997) adaptive lifecycle aims to reflect the “unpredictable realm of increasingly complex systems” (p 25). An iterative cyclical model, it attempts to improve on evolutionary approaches to development by not incorporating deterministic cause and effect rules. The first stage of the cycle is “speculation”. “In complex environments” he suggests “planning is a paradox...outcomes are unpredictable”. He considers planning a too deterministic word, laden with too much historical baggage. The second phase is “collaboration”, defined as a careful balance between management and control activities and emergent activities. After identifying which parts of a project are predictable, and therefore plannable, a project manager has to “establish an environment in which the wild and wonderful properties of emergence - basically open, collaborative, messy, exciting, diverse, anxiety-ridden, and emotion-laden - can exist” (p 26). The third phase of this iterative cycle is “learning”. This involves exposing products to a variety of stakeholders to ascertain the value of the products. The cycles need to be short, so that teams can learn from small, rather than large mistakes. He also suggests that they overlap and comments the phases are “purposely messy, non-linear, overlapping terms” (p 27).

As he suggests, this process model suggests “a terrifying prospect” for many. While this model does suggest chronological and hierarchical emergence, it is probably too innovative and different from existing approaches to managing the change process for most practitioners. Alternative process models should not only consider the context of development, but also the historical and organisational context of systems developers. Additionally, most practitioners are probably aware that the process is “messy”. A process model should provide a useable framework which can be used to structure and make sense of it and this approach does not do this.

4.4.3 Reflections: Viable alternative macro-level process models

Although the lifecycle model has come to be the almost unquestioned standard model of the process, there are a number of possibly viable alternatives, as described above. Several, such as Krogstie's system devtenence and Smith's V model, are insufficiently documented to evaluate fully. However, other models, such the participative STEPS approach, Eason's Multi-linear model and Rapid Application Development are possible alternative models.

Of course, the argument could be made that normative models are not always meant to be descriptively valid, but simply tools for justifying courses of action. This may be true, but does using models of process in such a way help us to improve the practice of developing appropriate and effective information systems in organisations? By deluding those who use them about the nature of the process and context of organisational change, they can focus entirely on the content of those proposed changes and remain only technical systems developers while their work involves non-technical issues. In doing so, they are not only applying what Kling (1987) would term a discrete-entity model of development, but enacting it. Kling pointed out that discrete entity models of context lead to overly ambitious projects. The dominant SDLC model could be simplistically but not unfairly called a discrete-entity model of process and can be linked to the current low rate of project success. An acceptable alternative model would recognise the closely intertwined nature of context, content and process by being descriptively as well as normatively valid. The alternative models identified here go part-way to fulfilling these needs.

However, these alternative models of systems development projects are unlikely to become common, at least in the foreseeable future. The lifecycle model has become so deeply embedded in organisational processes that to change them could be likened to a paradigm shift. One of the LSP project managers commented that these presented a radically different approach which scared her a little. She observed that project managers tend to prefer to change others, rather than themselves and noted that one of the necessary characteristics of a project manager was to be able to not consider anything that did not relate to the project at hand, at least during critical periods. Systems developers can also be held accountable for problems in projects, even if those problems are outside the scope of their control and using a recognised methodology can effectively illustrate they acted responsibly. In other words, adherence to a recognised methodology can offset personal criticisms. The lifecycle model is an integral part of most project management and system development methodologies and has a strong profelectic advantage over other approaches.

However, project management and system development methodologies have not just become widely utilised because they help justify individuals' actions and reduce personal accountability. They are also widely used because people find them useful. Although they generally have to be adapted to the specific context of use, they provide a useful starting and check point for considering the issues involved. They can also aid the development of common understandings of concepts and terminology within an organisation and can help manage the tendering process when developments are outsourced. Existing project management and system development methodologies are valuable tools for managing processes of change in organisations.

The problem is that projects are generally seen as the unit of change for systems development. This is problematic as there is a risk that change is viewed as a discrete event distinct and separable from its context (Pettigrew 1985). As Orlikowski (1988) suggested:

By focusing on the episode of change, rather than the process of change, attention is drawn away from the mechanisms through which changes occur and from which underlying patterns and continuity can be identified.

Although Pettigrew's work is almost 15 years old, his observations are of great relevance in the Tasmanian State Government today. Generally, visions for change are translated into single, monolithic projects. Such projects also tend to be initiated by executive management or the executive government and, being enforced from the top of the organisation, tend to encounter significant hierarchical emergence, which is often labelled "resistance" or reflective of a need for extensive "change management". Other current examples of such projects include Tasmania Police's *Project BATON* focusing on reengineering police work processes and the *Service Tasmania Project* aiming to develop a single shop-front for government services.

These projects are very difficult to manage due to their complexity. No wonder project managers are likely to rigidly stick to well-established methods! They are often unlikely to successfully complete such a project according to the well-established criteria for success in this area (completed on time and within budget and meeting the requirements of the stakeholders and the original objectives of the project). Adopting well-established methodologies is an effective way of offsetting any possible personal blame. In essence, they become "profelectic" managers.

In 1997 the CIPU's approach to managing projects became promoted as the standard approach in the State Service. Many people from other sections of the government have found their approaches and experiences valuable in pursuing a range of projects, most of which involve some computerised technology. However, in being promoted as "the answer" to managing change, their project management methodology runs the risk of becoming devalued as it is used for change initiatives which are simply too

large or complex to be defined as projects. In other words, there is a risk that the approach used by the CIPU is being used for purpose for which it is not suited. When these “projects” suffer problems, it may be the methodology itself, rather than its utilisation which is seriously questioned. In this way, the methodology’s value for projects where it could be of great use may be significantly undermined.

One possible solution is to recognise that initiatives for change do not necessarily have to be translated into a single project. Smyrk (1997) noted that change initiatives generally needed to be broken into a number of projects, one of which would possibly involve a significant IT component. Other related projects may include process reengineering or building projects. However, such projects may also be linked chronologically and occur as part of an ongoing stream of activities.

In other words, a change initiative could be viewed as a series of interlinked projects. In this way, chronological and hierarchical emergence do not have to be primarily managed within the scope of a single project. The challenge then is that the relationship between these projects and chronological and hierarchical emergence need to be effectively managed.

Essentially, there is a significant gap between strategic planning or policy planning and the level of individual projects and project management. Strategic and policy planning by definition continually look forwards to future requirements, possibilities and actions. Project management is concerned with implementing the strategic and policy plans of today and yesterday. What is required is another level of activity between strategic and policy planning to effectively coordinate the relationships between related projects.

At the moment, these tasks are done on a fairly ad hoc basis by project steering committees, senior and executive managers and/or by project leaders or project directors of large and cumbersome projects. This level of activity could be termed change program management. Within the scope of each project, the effects of chronological and hierarchical emergence would be limited effectively by their small size, and so could be managed according to well established approaches, which do not greatly recognise such emergence. Emergence is managed at the higher level of change program management.

In practice, change program management activities would combine elements of both project management and strategic planning. Risk management, stakeholder management and milestone and activity scheduling would be relevant project management techniques for this level of activity. Such activities could also incorporate the strategic planning concepts and as outlined by Mintzberg (1994). Program managers, or project directors would actively try and manage hierarchical

and chronological emergence, rather than controlling it and limiting its influence. They would be actively involved in negotiating and facilitating the definition of specific change projects in a given area. A project director would have to be a senior person who would be responsible for major decisions, ongoing emergent issues, relationships between projects and the link between projects and the strategic objectives they are to pursue. This approach has been adopted by the Service Tasmania Project and will probably be useful in other situations.

In conclusion, there are two primary ways to introduce chronological and hierarchical emergence at a macro level in practice:

1. adopt a different model of process, such as RAD or STEPS; or
2. embed existing processes based on the SDLC model within broader processes for managing change which do recognise chronological and hierarchical emergence.

Program management and the role of a project director are approaches to managing change in organisations that could be effectively linked with existing well-established system development and project management methodologies.

4.5 Social institution and action: A multi-level model of systems development consistent with Giddens' structuration theory

This dissertation introduces, justifies and describes a multi-level model of the process of systems development. Introduced in Part 2, the micro and macro levels of this model were discussed in Parts 3 and 4 respectively. This chapter considers the final institutional level of the change process and illustrates how the different levels fit together. Essentially, this institutional level of change is closely aligned with Giddens' structuration theory and illustrates how the process of change is deeply embedded in its organisational context. This chapter also illustrates that the three levels of the multi-level process model are tightly interlinked.

The theme of objectivism versus subjectivism, first raised when considering underlying axioms, pervades this thesis. This issue is not just an ontological one, but has practical significance as well, shaping how we view issues of planning, development and implementation. The debate between subjectivist and ontological stances is not only an academic one, but is reflected in many aspects of organisational life, management process and the systems development process. Simplistically, but not unfairly, this binary distinction between subjectivism and objectivism is identifiable throughout the issues raised in this project. This trend reflects observations by others, such as Hirschheim and Klein (1992) and Avison and Fitzgerald (1995) and echoes Burrell and Morgan's dichotomous division outlined in Part 1. This distinction has great relevance to the micro, macro and institutional levels of the process model described.

On the one hand, an objectivist ontological stance underlies the top-down approach to policy planning, change management, the systems development lifecycle model and much systems development literature. This stance is reflected in the language and concepts employed. Change is defined by senior managers who see it as an objective situation. Requirements are "analysed" and "determined" in the same way that a doctor examines a patient. Different groups involved have the chance to review requirements to check if they are "correct". Requirements can be determined and plans made at the beginning of what is seen as a largely linear, sequential process, with outcomes being judged according to how well they meet the plans defined at the beginning of the process. The high failure rate of information systems development is attributable to inadequate change management procedures: people need to accept change and be intellectually and emotionally ready for it. The intrinsic correctness of the changes is not in question and "resistance" is viewed as abnormal or illogical. The

changes need to be adequately communicated to them and the commitment of top management openly illustrated. In other words, the context is largely ignored or simplified, the content of the change is established up-front and the process is seen as a sequential one. The emphasis is on management's prerogative, the use of authority and expertise. Reflecting the criticisms made of the objectivist ontological stance, this top-down, rationalistic stance has been criticised throughout this project for not reflecting reality in complex organisations.

On the other hand, there is the subjectivist ontological stance which recognises that reality is largely socially constructed, and social processes surrounding systems development have a large impact on the way the process of systems development unfolds. The context is not wholly discernible at the beginning of the process and is not stable anyway as it is being continually renegotiated through social interactions. Thus firm long-term plans for implementing specific major changes are very difficult to achieve and there is a close relationship between planning and implementation. With its emphasis on negotiated, subjective reality, there is an emphasis on participation, industrial democracy and hierarchically more bottom-up processes of decision making.

Vidgen and McMaster (1995) comment that the ontological division between objectivism and subjectivism pervades IS development methodologies and has led to a "seemingly implacable dualism of scientism and interpretivism". While such alternative approaches emphasise the need to consider the context of the development process, they seem to ignore the context of the systems developers and others. The binary division between subjectivism and objectivism is useful for analytical purposes, but hides some important issues. The application of authority and expertise plus the constraining and enabling effect of institutional elements is not addressed. Giddens' structuration theory provides an alternative, which underlies an emergent approach. The institutional level of the process model is based on structuration theory and focuses on the dualistic relationship between objective and subjective elements.

4.5.1 Chronological and hierarchical structuration

If people see a situation as real, then it is real in its consequences (Merton 1968). Despite its many problems, the top-down, rationalistic, objectivist approach is the norm in bureaucratic organisations such as the Tasmanian State Service. For many, these norms are an objective reality which structure their operations. These norms may be a social creation, but for many there is no choice but to follow them. They are as real in their consequences as physical furniture and, if people challenge them, the consequences can be as painful as bumping into the corner of a table. The need for changes such as those involved with the LSP may be socially created, but for

members of the CIPU and OPC, they were largely as real as the buildings in which they operated. Issues were presented as an objective fact and legitimised through authority and expertise. They were not negotiable. The senior manager responsible for the OPC emphasised that the executive government *required* that there be automatic consolidation. In other words, the position people have in the organisational hierarchy can shape the way they view their immediate social surroundings.

Again, once decisions are made and agreed to generally, they become as objective a reality as decisions enforced by authority and expertise. Negotiated reality can be challenged through interaction and created over time as people interact and come to agreements of meaning and interest. Later in the LSP project, the broad aims of the project were largely objective and set in concrete while previously they had been the subject of a great deal of negotiation. Thus, chronologically, objective reality can be created in the eyes of those involved through the negotiation of meaning and interest as well as the use of authority and expertise.

In summary, a subjectivist view emphasises the actions people take and the choices they can make. The implications for processes, such as systems development, is that it is likely to be at least partly a bottom-up process, often with no pattern to its progress. The associated risk is that changes lack direction in terms of the organisation's purpose and when change does occur, it is usually only incremental. The objectivist view emphasises the role of structures (through management's prerogative) and a lack of choice (change is mandated by the environment). The implications for the process of systems development is that it is likely to be top-down and linear. The associated risk is that over-ambitious and unrealistic plans are likely to be made, with the resulting failures attributed to change management problems.

Chronological and hierarchical emergence challenges the incommensurability of these two stances. Iivari and Hirschheim (1996) saw that different stances could be appropriately adopted in relation to different types of systems. Here the argument is that the different stances occur within the one systems development project. Giddens' structuration theory provides a key to the relationship between objectivism and subjectivism and forms the basis of the emergent approach. Thus, this dualism operates not only at the ontological level, but in practice as well.

What a person sees as an objective structure depends on their role in relation to the organisational context and the point of time in question as well as their personal attributes, and this is a crucial issue when considering any process of change. For some, such as the OPC's administrative assistants, management structures form part of the objective context/structure in which they work. Others, such as the chief

drafter, help to create such management structures through the decisions he makes and the actions he takes. What for some is an objective structure which frames their actions, is the realm of action for others.

This is what Giddens referred to as the duality of structure. This dualism does not just occur chronologically, as is suggested by Orlikowski's (1991) paper, but hierarchically as well. In other words, what is the realm of action for some is a structural frame for others which shapes their action. This is perhaps one reason why structuration theory is so difficult to apply to case studies and is a crucial issue which underlies the process of change²⁶.

Aspects of context can structure people actions hierarchically and/or chronologically. A plan structures our actions chronologically (Mintzberg 1994) while enacting one's role in an organisational context is structured not so much by previous decisions but perceptions of hierarchy and authority structures. The issue of managerial control/standardisation and industrial democracy/empowerment helps to illustrate this point. Standardisation, such as that associated with the LSP, can structure people's actions both hierarchically and chronologically. If it is created democratically through consensus decision-making, standards structure people's actions chronologically, but not hierarchically. If standards are enforced by continual management intervention and control, then actions are being structured hierarchically.

4.5.2 Structural frames and realms of action

People's realm of action is governed by their choice, so constitutes part of their subjective world. However, it is bound by structural frames which they perceive as an objective fact. For example, the precise way that I have written this thesis, the detailed concepts and language I have employed has been my choice – my realm of subjective action. I have never questioned the fact that it is in English, as for me this is an objective fact. As Berger and Luckman (1966) commented,

As a sign system, language has the quality of objectivity. I encounter language as a facticity external to myself and it is coercive in its effects on me (p 53).

In the same way, the desires of the elected executive government for better processes for producing legislation and better access to it and the vision of the senior manager for automatic consolidation was as real to members of the OPC and CIPU as the fact I would write this dissertation in English. The CIPU were given some firm goals to meet, but how they met them were more up to them. These goals were objective to them and would be the basis by which their actions would be judged but the way in which they met these objectives were largely within their realm of action. Embedded

²⁶ That is, that people's actions are structured by different elements of the context, depending on their position.

in the structural frame are the imperatives of the context which mediate choice and action.

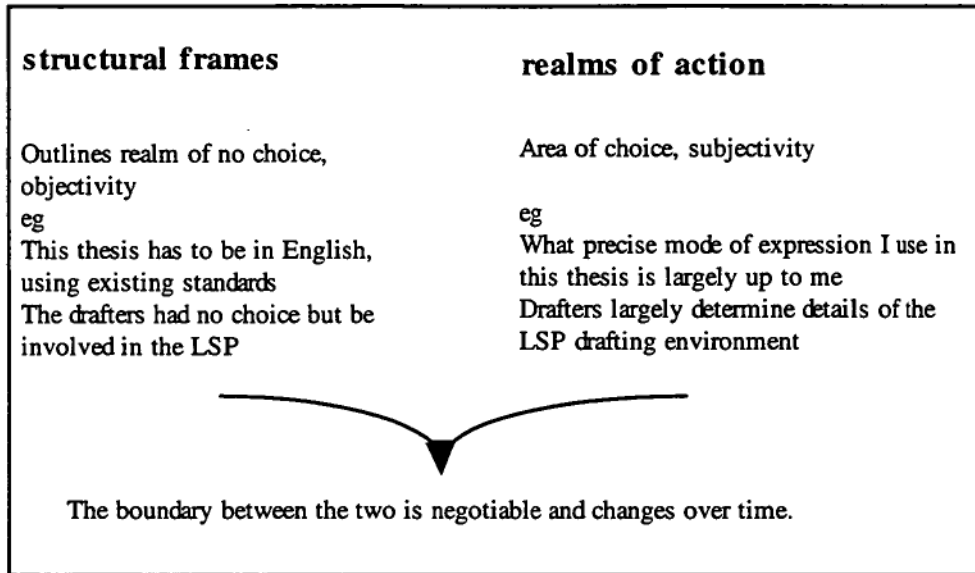


Figure 4.5.1: Structural frames and realms of action

This frame is not stable and is continually renegotiated through social interactions and so changes over time, systems development providing a good example of such changes. During the course of the LSP, for example, the goals of the project were renegotiated slightly, so that the diffusion of the legislation database was not included, with the agreement of the project's steering committee. Such decisions were within the realm of action for the steering committee as a whole, but were a structural frame for members of the CIPU and OPC. However, the steering committee would not have made such a decision without advice from the CIPU and OPC.

The systems development lifecycle, and the methodologies associated with it, significantly, were given an objective status. That is, they were perceived as reality, rather than interpretations of reality. For example, when asked what they were doing and why in relation to the creation of the functional requirements document and other activities, the systems developers referred back to methodologies to justify their actions and give them a rational, objective status. In this way, the objectivist concepts reflected in these approaches became tools for negotiation as people used them to justify their actions, a trend also noted by Buchanan and Boddy (1992). External issues, such as the dissatisfaction of parliamentarians, were commonly used to justify actions. Presented like tablets of commandments handed down from above, these external realities were presented as non-negotiable. The application of perceived expertise and authority aided this process and hierarchically structured people's actions.

The actions of the CIPU and OPC were limited by authority, expertise and physical resources. For example, the OPC were informed they would have to change some of their practices and utilise computerised technology. They had no choice: the authority of the senior manager and the executive government made this an issue beyond their realm of choice and helped to structure their actions, as did the perceived expertise of the CIPU project team, which further legitimised these decisions. Members of the OPC and other commented numerous times that they had no choice but to accept these issues. Yet members of the OPC had latitude in deciding the details of these changes and implementing them. These issues were actively negotiated and were part of OPC's realm of action. When negotiating these issues, viewpoints were constantly justified with reference to objective, external issues.

These observations illustrate how people's actions are structured and, in doing so, helped build a descriptive model of the process of planned change in organisations at an institutional level. This descriptive model incorporates chronological and hierarchical emergence by recognising that changes are deeply embedded in their context. An effective normative process model will explicitly recognise the structuring and structured nature of the organisational context of change and its influence over the actions of the people involved.

4.5.3 Changing boundaries between the realms of action and structural frames

Writers on change often emphasise that it is necessitated by a need to adapt to a changing environment and then mention the many ways the world has changed through reference to issues such as globalisation, increased awareness of environmental issues, increased competition and so forth (for example, Johnston et al 1996). What they rarely examine is the way in which these trends influence internal organisational changes. The need for change is generally described in objective terms with reference to the prerogatives of the environment. The senior manager promoting the LSP emphasised that the executive government *required* that the LSP go ahead. People continually emphasised that they had no choice but to change – the changes were enforced upon them.

As suggested by Actor Network Theory, objects are created which signify socially negotiated meanings via social interactions and, through their reproduction, they become legitimised and embedded in their organisational context (Latour 1995; Vidgen and McMaster 1995). In doing so, they take on the attributes of what Latour would term a "black box". These objectified, but socially negotiated concepts structure future social interactions. Technological applications provide a good example of such a structural frame. Specific technological applications can embody

theories about the nature and organisation of work (Markus and Bjørn-Anderson 1987). As Smyrk (1997) commented, when purchasing an IT package, one is primarily obtaining the processes embedded in that technology: the technology is merely an adjunct to it. Several drafters commented that the EnAct system reflected assumptions of the systems developers, rather than their approach to drafting.

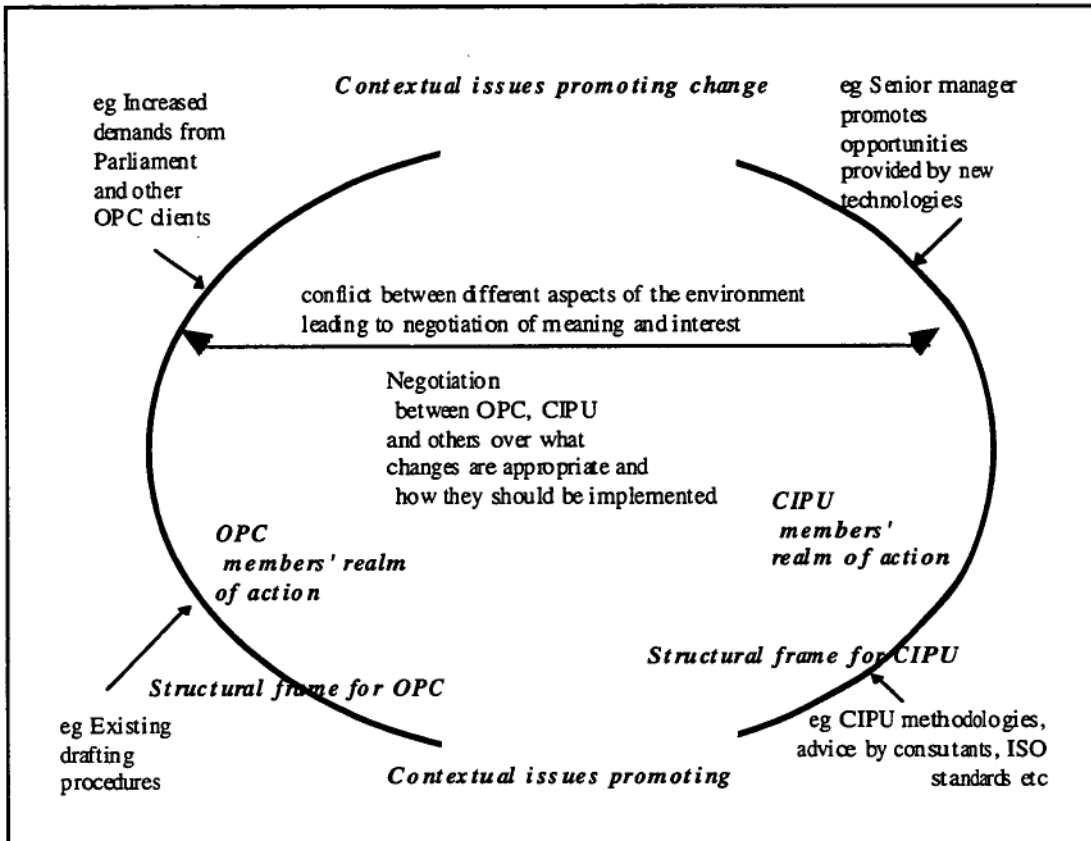


Figure 4.5.2a: Conflicting contextual issues, structural frames and realm of action for members of the CIPU and OPC in the early stages of the LSP

Changes in the (objective) context of an organisation promote changes within the organisation because they conflict with the (objective) established norms of the organisation. Thus there is justification for both sides of a debate in interest and meaning and the opportunity to recreate the social order. In this way, the increasing amount of legislation being created by Parliament, tight resources, an increasing perception that new computerised technology could be beneficial, collided with existing drafting and management processes in the OPC. These competing objective realities provided a framework in which to debate what changes would be appropriate to resolve the anomaly between them. This is illustrated graphically in Figure 4.5.2a.

Over time these conflicts were resolved and the boundary between the structure and realm of action changed via the negotiation of interest and meaning plus the use of authority and expertise. Later on, as is illustrated in Figure 4.5.2b, the signed off functional requirements produced as a result of negotiations early on structured participants' involvement and promoted stability, as did ongoing reviews by the

steering committee. However, contextual circumstances were changing slightly and, combined with increased knowledge about both technical and drafting issues, there were forces promoting changes and continuing debate. These debates occurred through the ongoing use of prototypes, the creation of detailed design requirements and testing procedures.

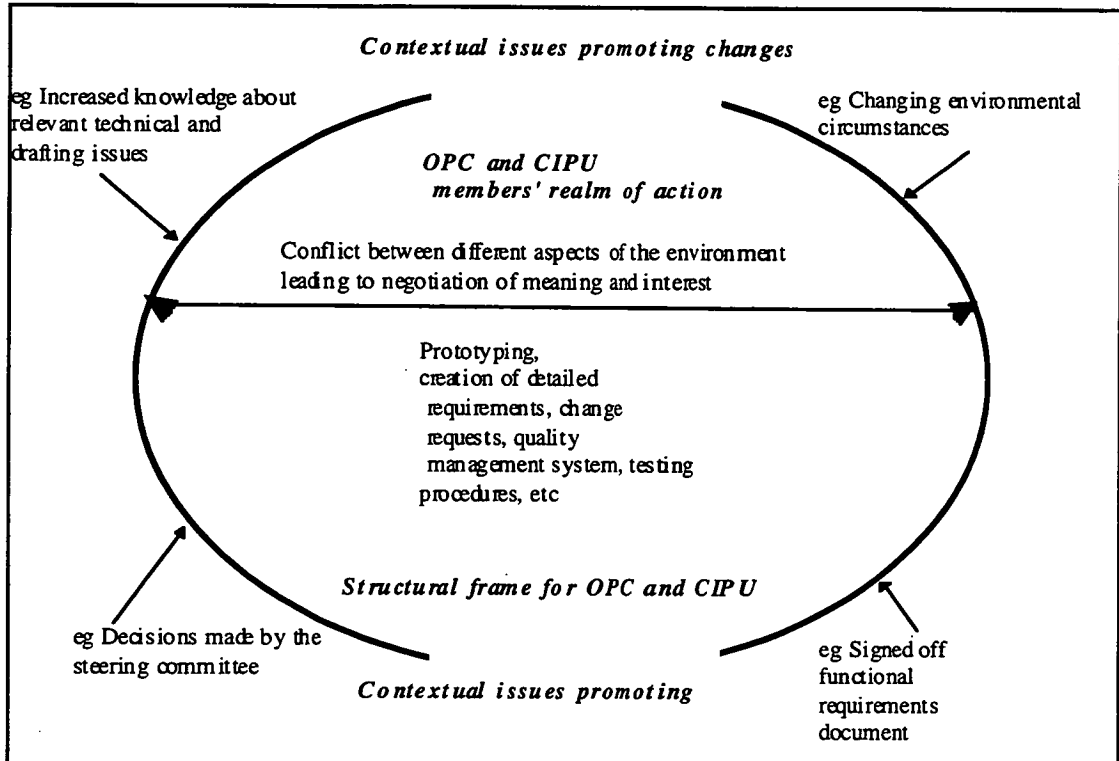


Figure 4.5.2b: Conflicting contextual issues, structural frames and realm of action for members of the CIPU and OPC in the early stages of the LSP

Thus boundaries between perceived subjective and objective worlds changed throughout the course of the project. Figures 4.5.2a and b provide snapshot models of frames of structure and realms of action for members of the OPC and CIPU generally. Figure 4.5.3 illustrates the movement between these realms of subjectivity and objectivity over time in relation to these models. These models provide just one example of how there can be movement in the boundary between the context and its structuring effect and people's realm of action. For other groups, sub-groups and individuals, the frames drawn would be slightly different.

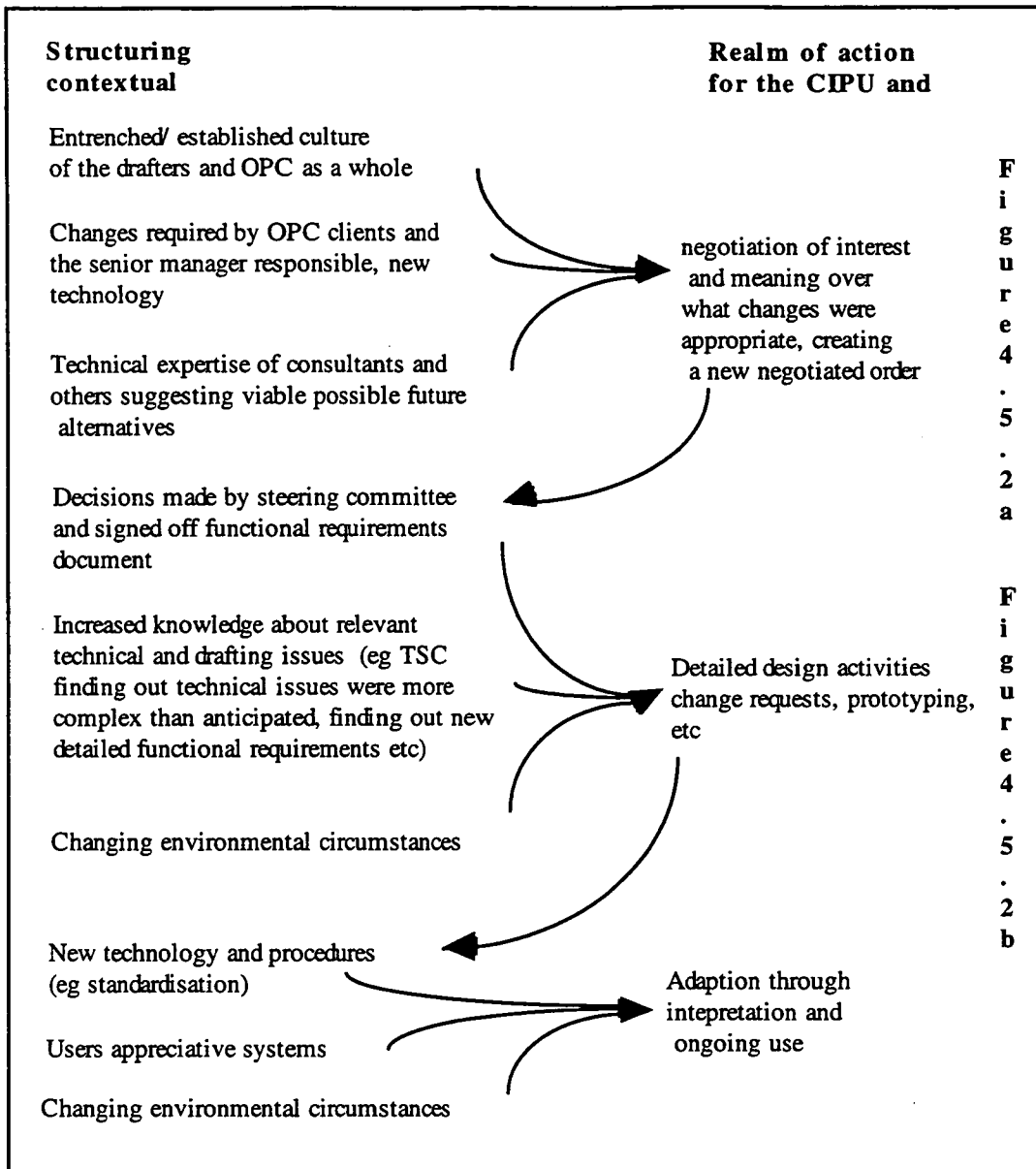


Figure 4.5.3: Interaction over time between realms of action and structural frames during the LSP

People's objective realities then provided a starting point for negotiating change and were used as resources in the negotiation process. The drafters attempted to objectify their working practices by likening it to a creative craft and emphasising their long experience and wide knowledge in the area. According to the senior manager, the OPC had to change because of the demands of Cabinet, Parliament, and the people of Tasmania. Given public service conventions, how could members of the OPC oppose such an strategy? The active support of top management is often emphasised because it provides a good resource for objectifying issues and imbuing them with authority so they can acquire legitimacy and debate can be sidestepped.

This collided with the established conventions and culture of the OPC but, combined with the technical expertise providing viable future scenarios, the push for these

desired changes proved effective negotiation tools. Once defined, the LSP project became a firm, objective reality to members of the OPC and CIPU through the decisions made by the steering committee and the signed-off functional requirements document (amongst other elements). These decisions and document provided an objective reality which guided people's actions but sometimes conflicted with information and understandings developed through ongoing interaction between all parties and greater exposure to relevant details. Along with some environmental changes further negotiation of detailed issues were promoted. Thus the technology becomes adapted through its interpretation and ongoing use.

In other words, people act according to their own perceptions but only within a framework of possibilities defined by the historical (chronological) and organisational (hierarchical) context. The different perspectives of users in an information systems development project are crucially important to the success of that project. Yet these opinions are only relevant and acceptable if they are aligned with the overall goals of the organisation and their role in the organisation. For example, a public servant involved in a systems development project that has been initiated by executive management and the elected government is obliged to follow their directives whether they agree with them or not, in principal at least. As several drafters commented as the system was implemented, they realised they had no choice but to use the system, given the investment in it and the advantages to the wider community. They can dispute the manner in which the directives are implemented but, while they may privately disagree with these directives, to a large degree they are just that: directives that are to them an objective fact. This is the hierarchical structuring of actions.

When decisions are made, there may be an element of choice as concerned parties are consulted and involved, but once the decision has been made, the decision becomes something very close to an objective fact for those involved or affected by the decision. It may be open to dispute or interpretation, but to a large degree, decisions objectify the world of the people affected by them. Hence, later in the project, the functional requirements document was treated as an objective fact by which to judge the progress of the rest of the project. This is the chronological structuring of action, and the link between this and hierarchical structuring is obvious, as decisions cannot be enacted without some authoritative backing.

This movement between subjectivism and objectivism is related to Lewin's concept of unfreezing and refreezing an organisation before and after a change. The difference here is that this unfreezing and refreezing is viewed as an integral and ongoing part of the change process and not just adjuncts to it and the unfreeze/ change/ refreeze model suggests that change occurs in a continuum of stability. This process of structuring actions chronologically was examined by (Berger and Luckman 1966) in

depth, who referred to it as the process of objectification. Issues can become objectified through processes of social negotiation. They can also become legitimised through authority and expertise. Berger and Luckman referred to legitimation as "a second order objectification of meaning" (p 110). Giddens recognised that these processes occur simultaneously, while Berger and Luckman look at the processes occurring chronologically (Willmott 1993). This analysis of the LSP backs up Giddens' theoretical arguments.

However, there is little investigation on how such processes occur hierarchically, as investigated here. This process of hierarchical structuration occurs through negotiation and the application of expertise and authority. Berger and Luckman(1966) pointed out that knowledge is one key area of this dialectical process as it helps to externalise issues and presents the world in an objective manner, hence the importance of perceived expertise in implementing change.

The process of objectification includes three crucial moments in the process of creating an objective reality (Berger and Luckman 1966). Externalisation occurs when products of human activity (eg negotiated outcomes) are given an objective character independent of those who created it (eg in the LSP, systems development methodologies are said to be the reason for undertaking certain actions).

Internalisation is the "intersubjective sedimentation" of these issues, when a previously continually negotiated issue becomes embedded in some sign system (for example, the Quality Management System used in the LSP). Reification occurs when such products of human activity are treated as if they were something other than human products (in the LSP, for example, drafters strongly opposed any changes which could impact on the broader processes of producing legislation and especially parliamentary processes). Berger and Luckman only really look at the process of objectification, but the opposite also occurs during change. Externalised, internalised and perhaps even reified beliefs can be challenged and renegotiated. No wonder change can be traumatic!

Change is movement. Movement means friction. Only in the frictionless vacuum of a non-existent abstract world can movement or change occur without that abrasive friction of conflict (Alinsky 1971 in Andrews 1993).

When the senior manager and others promoted the LSP, established interests and meanings were threatened. This situation provided an opportunity for negotiating change as previously objectified issues were subjectified and brought into the realm of choice and action. Such a situation is likely to be a challenge to all involved and was. Decisions were the end product of heavy negotiating by all involved.

4.5.4 The creation of a new negotiated order

Later, these decisions became institutionalised, unquestioned and objective. People did not question the use of computers generally in the OPC to anywhere near the same degree in the later stages of the project compared with the earlier phases. The signed-off functional requirements document became a largely objective measure which guided (structured) future actions. On the whole, wider issues, such as the worth of the project generally, were negotiated earlier in the process, while later negotiations revolved around more detailed issues (such as standard amendment wordings). One could imagine that in an unsatisfactory systems development process, the negotiation of broader issues may be left to the later stages, or not at all. In this way, a successful systems development process is the creation of a new negotiated order.

Thus, the boundaries were established not only by the social positioning of those involved, but changed over time. People's realms of action extended and contracted during the course of the project. The early part of the process involved subjectification, as established social structures were questioned and were negotiated. Later on, though, as changes were negotiated, they regained a more objective status as they became embedded into the organisational context. This can be described as a process of objectification.

Much of the time changes in these boundaries seemed to occur when conflicts occurred in the structural context which shaped people's actions. People would start to disagree on what should be done and the resulting conflict resulted in negotiations of interest and meaning and change. The development of a prototype by one of the users provides a good example of this.

Social systems, such as organisations, are essentially bodies of negotiated order. Inconsistencies exist - human groups are never homogenous in all respects – but organisations work as a body via negotiated meanings and interests embedded in agreed upon language and communication structures, concepts, power structures and networks of legitimate norms. Through the interaction of organisational members with conflicting opinions and interests, these structures of signification, domination and legitimation are renegotiated to form a new social order.

The model described above is a description of change at an institutional level. The concepts of hierarchical and chronological structuration, the difference between structural frames and realms of action and the changing boundaries between the two provide an abstract explanation for how changes occurred throughout the LSP at an institutional level of analysis.

4.5.6 Reflections: A multi-level process model

It is a truism to say that information systems development is a process of change in organisations, but it is an important point. As Pettigrew (1985) pointed out, change can usefully be viewed as three fundamental and intertwined elements of context, content and process. While the focus of my research has been on the process, these other two elements must be considered if we are to understand it.

The context of change basically includes elements and issues which impact on or are impacted by the content and process of the change. The descriptions of the LSP's broad context reveals that, while the LSP is a unique project in that it combines both information retrieval and drafting facilities, the trends promoting it are similar to those in other jurisdictions. The immediate context of the LSP includes government agencies and, particularly, the Tasmanian Office of Parliamentary Counsel (OPC). Important contextual elements include:

- the purpose of the organisation and place in the wider context (the purpose of the OPC is to create legislation and maintain Tasmania's statute books);
- the technical and process infrastructure (ie. the processes for creating and maintaining the statute book and enabling access to it);
- roles (in the case of the LSP and OPC, those of drafters and their support staff particularly);
- authority structures; and
- culture.

These elements all had a great impact on the content and process of the LSP. The OPC's purpose helped shape the resulting EnAct system, while the largely bureaucratic authority structures and the stable organisational culture combined with a lack of past experience in computerised technology and the highly articulate people who made up the OPC are just some of the issues which helped shape the process.

Not only do contextual issues shape the content and process of change, but even our perceptions of the context seem to greatly influence how we perceive the process should evolve. Kling (1987) effectively illustrated that when systems development initiatives are viewed as a discrete entity from their surroundings, highly ambitious projects can be defined and are seen to be achievable through a process which does not take into consideration the continually changing and socially created nature of the organisational context. Alternatively, viewing a planned system as part of an integrated web results in far less ambitious but more realistic projects and the process of achieving them is viewed as largely a social and political one. Checkland's soft systems methodology focuses on the differing perspectives people have of the

situation and the systems development is portrayed as a learning process. The model of the organisational context described above was influenced by the work of Kling and Checkland, amongst others, and this has an impact on how the process is portrayed here.

The discrete entity approach described by Kling underlies the commonly used structured approach to systems development. One result of using this structured approach is that contextual issues are largely ignored until the implementation stages of a systems development project, where they are sometimes labelled as issues of “change management”. As argued earlier, this phrase “change management” is problematic because it can refer to at least three different types of activity. The interpretation used with this structured approach is problematic in that it simply aims to make “them” change to align with “our” view, without “them” having meaningful input into what these changes involve. As such, it reflects highly rationalistic managerial assumptions. Alternative conceptions of change management do not reflect these assumptions, but this interpretation is commonly employed *within* systems development projects (in which the *content* of the proposed changes have basically been established). All three interpretations were used within the LSP, though this rationalistic one dominated.

However, the users did have a significant input into the content and process of the project, though not always in a way the systems developers anticipated. The initial response of the primary users (the drafters and their support staff) to early prototypes and the functional requirements was largely negative. Their actions and opinions were labelled as “resistance”, or as indicative of a lack of understanding of the situation, the technology or both. These are both typical phrases and concepts from the literature and standard systems development approaches.

To understand this approach, there is also a need to look at the context of the systems developers, as well as the users. The primary unit responsible for the LSP project, the CIPU put a great deal of effort into aligning with recognised standards of best practice and typically their actions were aligned with standard systems development approaches. Thus criticisms of their approach should not be viewed as criticisms of the people involved. They were only aiming to apply commonly recognised standards, but these standards were not always adequate. The fact that the CIPU aimed to adopt standard practices also suggests that observations from this project may have significant implications for other projects using these approaches.

Essentially, while systems and their application within organisations have become increasingly sophisticated, our conceptions of the process by which they are achieved have lagged behind and are causing some problems. EnAct will probably be

considered a satisfactory system in the long-term, but the process by which it was achieved suffered major delays and user acceptance issues were an ongoing (though ultimately unrealised) concern. Even though they formed part of the same department, the systems developers' actions were structured by a quite different context than the members of the OPC and their differing frames of reference can be compared and contrasted. While the OPC can be classed as a stable, bureaucratic unit comprising of people who worked individually, the CIPU had a technocratic orientation, a team based approach to work and roles and tasks were quite fluid and dynamic.

These differences greatly shaped the way the LSP unfolded and the resulting EnAct system. The process underlying the LSP involved a great deal of negotiation and learning by all involved. Social processes which facilitated this included:

- *active* involvement by participants;
- the negotiation of meaning and interest;
- the creation and sustainment of coalitions of meaning and interest; and
- the application of authority and expertise.

The in-depth examination of these processes as they occurred throughout the LSP in Part 3 adds to observations by Walsham (1993) and others who have focused on the social processes surrounding systems development. At one level, the process of change can be viewed as social interaction.

Process refers to movement, or activities over time. Thus conceptions of temporality are important. Giddens (1979) suggested there are three forms of temporality:

- *durée* (micro, social processes- as discussed above);
- *dasien* (or the lifecycle of the organism- in this case, the system; macro processes); and
- *longue durée* (or institutional time).

A major contribution of this dissertation is the focus on these three levels of temporality in relation to empirical observations. These observations illustrate the closely interconnected nature of these three forms of temporality and suggest that the commonly used model of process operating at the *dasien* level (the systems development lifecycle), is not compatible with the manner the process unfolds on the other two levels. Although widely criticised, the systems development lifecycle is widely used and was formally adopted during the LSP.

An analysis of the activities involved in the process of the LSP suggest it was far more emergent than the common model implies. This emergence occurred both

chronologically and hierarchically. In other words, the project emerged over time and was the result of the interaction between different people involved, rather than just some. The micro, social processes are the engine that drives the more macro processes and clearly are emergent as social action unfolds over time. The process at an institutional level is also emergent as institutions are created and recreated gradually over time through ongoing social interaction. These three levels are all tightly interconnected and so the macro level of temporality should also reflect emergence.

In practice, at the macro level, emergence entails:

- participation;
- incrementalism; and
- iteration.

This emergent approach aligns with Mintzberg's (1994) approach to strategic planning, Giddens' (1979) theory of structuration, Quinn's (1989) concept of logical incrementalism, Kling's (1987) ideas about the negotiated nature of computing developments, Truex's (1993) concept of emergent organisations, and Orlikowski's (1996, 1997) theory of improvisational change, but not a great deal of the systems development literature, which only tend to only pay lip-service to these practical applications of emergence. However, this dissertation identified a number of alternative process models which do reflect these crucial elements of emergence, the most promising being: STEPS, an elaboration of Mumford's ETHICS methodology, Eason's multi-level process model and particularly Rapid Application Development (RAD). Alternatively, the emergent nature of systems development can be recognised by not viewing significant systems developments as "a project", but as a series of small projects linked together as a change program or under the care of a project director. The scope of and relationship between these projects should be carefully managed. This level of activities would include elements of both project management and strategic planning, but would not be precisely defined due to its emergent nature.

At the third, macro-level of institutional time, this research echoes that of Orlikowski and Hofman (1997) and Barley and Tolbert (1997) who have illustrated how the process of structuration unfolds over time. The core element of structuration theory – the dualistic relationship between structure and action, or objective elements and subjective ones – pervades this thesis and underpins all levels of temporality. At the micro, social level, culture, authority structures and roles form structures of signification, domination and legitimation. People's actions unfold as they negotiate meaning and interest via communication, power and sanction. At the middle, dasien level, established models, such as the systems development lifecycle, structure

people's actions, but these models are also created and enacted by them. Structuration theory adequately illustrates that the process of change must be viewed as chronologically emergent.

However, it is difficult to apply in practice, at least partly because people perceive structure according to their position and perceptions. In this way, structuration not only unfolds chronologically, as Orlikowski and Hofman and Barley effectively illustrated, but hierarchically as well. Acting within their own frames of reference, people's actions are structured by differing elements. What for some is an objective structure which frames their actions, is the realm of action for others. For example, the decisions of the chief drafter were at least partly in his realm of action, while they formed the objective structure for those affected by them.

One implication of this is that, despite its many problems, the top-down rationalistic approach, as embedded in the systems development lifecycle, is the norm in organisations such as the Tasmanian State Service. People are required by top management, tendering requirements, public service conventions and audit expectations to follow them. Despite its many recognised problems, it is embedded in such organisations, and any alternatives would have to either align with it or represent a paradigm change.

RAD or the articulation of systems development initiatives as a series of intertwined project are possible alternative approaches identified here. Such alternative macro-level process models would recognise the socially negotiated nature of change, the close connection between the planning and implementation of change and the ongoing influence of the organisational context. They would also be aligned with the micro and institutional levels of the process model described here.

5 Reflections and Implications

The process of drafting legislation can be compared with the process of creating an information system. Certainly the outputs are different, but the outcomes of both are intended to structure future actions. The similarity between the two processes has become apparent on a number of occasions. Lim (1993) defines legislation drafting as "legislative engineering" in much the same way that some systems development activities are often referred to as "software engineering" and systems development tools such as flowcharts have been used by Commonwealth drafters to aid understanding (Australia 1993). Friedman and Cornford (1989), for example, likened systems development to a creative process of writing which could not be strictly prescribed as a series of linear steps. The drafters have to obtain the policy issues from their instructing officers in much the same way that systems analysts have to "determine requirements" from users. The drafters, though, seemed to recognise that policy issues were often not completely defined when the legislation was being drafted, while the systems developers, in adopting the assumptions of their normative literature, assumed that user requirements were fixed "facts" to be gathered.

The interesting difference between the two approaches is that, while systems developers emphasise technical and procedural efficiency and tend to ignore political issues and social interaction, the drafters do not. In fact, the systems developers and some others might argue that the drafters have an inverse emphasis to the systems developers. The drafters emphasise the political nature of their processes and recognise that expertise of instructing officers and the hierarchical authority of politicians are likely to change the outputs of the legislation production process. The process of creating a piece of legislation is seen as an iterative and emergent one and ongoing changes are seen as an integral part of the work. This is in contrast to how systems developers tend to view their work processes. Systems are defined in plans and then implemented, with changes made to previously established plans being a result of unforeseen problems. No wonder systems development projects are seen to have failed so often (Clegg et al 1996).

A comparison between the two different work processes helps us to understand some of the problems the systems developers had in comprehending the work of the drafters. Perhaps one of the major reasons the systems developers had difficulty in understanding the worldview of the drafters and their support staff was that they tended to view the drafting process in terms of their own assumptions about how work processes were or should be completed. They tended to assume that the process of drafting was a top-down one, in which the structure could be defined before the content was written, in much the same way the process of systems development is

commonly envisaged. In essence, they imposed their world view on the users as they tried to develop requirements for the system. Many of the negotiations surrounding the project centred on this interpretation, as many of the drafters questioned the systems developers' interpretations of their work.

Drafters have to recognise that changes will often happen while they are writing legislation. Systems developers often do not and tend to assume the policies being articulated at the beginning of the systems development process will remain stable for at least the course of the project. Our whole project management and systems development approach is based on this assumption, but how realistic is it?

It is a common experience for OPC drafters to have to deal with changes in policy while a Bill is being drafted. New material is added, existing policy decisions are changed, and other policy decisions are dropped entirely. Naturally, these changes have a very bad effect on the quality of the final product... it is like having to build a sports car, and then, while you are building it, being told to turn it into a sedan, and then being told to turn it into a bus. This is no way to win a design award

...Another important factor is that when Acts are amended over and over again they lose their original design and become more and more complex through accumulated additions and modifications (Melham, 1993: p 7-8).

Perhaps this alternative approach to work processes – one which recognises the implications of social interactions – can provide some suggestions for improving the process of systems development. There are indications that the process, as formally conceived by the systems developers early in the project and in the systems development literature generally, also did not adequately provide a description of how events unfolded. Some would argue that the approach of some of the drafters emphasises social interaction at the expense of efficiency and effectiveness, but perhaps some insights can be gained from such an alternative approach.

It could be argued that the legislation drafters operate in a very political context, while systems developers operate in a more technical one. Yet is this the case? Walsham (1993) illustrated that an insight into the process of developing information systems could be achieved through a political and cultural analysis of the situation and Part 3 has provided what is very close to Walsham's political and cultural analysis of the situation, an analysis he describes as a process model.

While not disputing such analyses usefully give an insight into the dynamics of systems development at a micro-level, they are not process models as they do not give a sense of movement over time. If we liken change to a journey through time from one state to another, then a cultural/political model is like the motor of our vehicle: it is the means by which we travel and the reason for our locomotion. It does not, though show us a map of the journey itself.

The observations from this research project stand in direct contrast to the literature calling for radical organisational changes alongside the implementation of new

technologies. As the first LSP project director commented just before he retired in early 1997,

The LSP was an incredibly risky project. In hindsight and 20/20 vision - and hindsight is a good thing - it probably would have been easier to have implemented a smaller scale project. For example, we could have just implemented wordprocessing technology. As it was, we went from the stone age to the space age in one hit. There has been a lot of talk about introducing radical or revolutionary change in the literature... (interview 10/3/97).

There have been many comments on the perceived low success rate of systems development projects. The problem may lie with the inadequate way we, academics and practitioners involved with systems development projects, conceive the process of development and implementation.

The observations here should in no way be interpreted as critical of those directly involved in the LSP. They aimed to utilise recognised standards of best practice, and it is inadequacies in these standard models which can be linked with problems with the LSP. The process of drafting legislation can provide an illustrative or comparable process which recognises chronological and hierarchical emergence to a degree.

5.1 Summary of Conclusions

The empirical focus of this project has been the Legislation Systems Project. Involving the core government processes of legislation drafting and enactment, a project such as the LSP could potentially have an impact on not only the manner in which legislation is drafted, but on parliamentary processes. However, due to the possible political ramifications, the systems developers and others involved in the project have aimed to reduce any immediate effects on the enactment of legislation by generally treating these processes as a given, objective fact. While computerisation may induce changes in the workings of parliament and the processes of enacting legislation in the longer term²⁷, it is unlikely to occur directly as a result of a single project such as the LSP. However, the project could feasibly have a great direct impact on the processes of drafting legislation, and this is the primary empirical focus of this research.

The LSP was initiated and strongly supported by both parliamentary and executive government top management, but was not greatly or widely embraced by the OPC. The project also involved the development and use of sophisticated and novel technologies in a complex organisational context. The reactions of the members of the OPC to the resulting EnAct system have ranged from a full and satisfied acceptance through to a belief the system was acceptable, but needed some fine

²⁷ Though probably not on quite the scale envisaged by Guy Fawkes.

tuning, to a stance that the system was not yet achieving the promised benefits and open dissatisfaction with the results of the four year project. The wide diversity of opinions within the OPC precludes any attempts to firmly judge resulting system a success or otherwise, and this will be determined in the longer term as the OPC utilise the technology. However, the active utilisation of the system by the majority of staff members and their general acceptance suggests that, in the longer term, the EnAct system will be considered successful. Certainly, the system has promoted interest from other OPC's and most are impressed at EnAct's technical sophistication.

However, the process of developing the system cannot be considered a total success. There have been substantial delays and user acceptance remained an ongoing and justified concern. These problems can be traced back to the manner in which the project was planned and defined. To attribute them to factors such as "the use of novel technology" and "a need to educate the users" would be trite and unhelpful for future situations. Such problems can be traced back to how the process of systems development is viewed.

A culture of technocratic rationality pervades the area of information systems development, both in terms of how the process of development is portrayed and the manner in which the organisational context of the computerised technology is to be managed. Thus, the ontological and epistemological issues addressed in Part 2 of this dissertation have relevance to practitioners as well as researchers. This research project reflects an interpretivist epistemological stance and is critical of this dominant positivism in both research and practice.

This research has employed Giddens' structuration theory as an ontological sensitising device for examining the process of systems development. On commencing this project in 1993, I aimed to track the process of technologically induced change that was to begin with the LSP. However, with the ongoing delays in the project, I changed focus to the process of developing and implementing such a project. Such a change is acceptable and not unusual when studying phenomena in their organisational setting as changing circumstances may require it (Nandakumar and Jones 1997).

I have used Pettigrew's (1985) contextualist analysis as a framework for analysis. The dissertation is structured around the three fundamental elements of change Pettigrew identified. Part 2 focused on the context of change while Part 3 examined the content of change and how has been established and Part 4 looked at the process of developing information systems given the issues covered in the previous sections. The dissertation was structured to reflect the iterative and emergent nature of both this process and the process of conducting research. Due to this iterative and emergent

structure, its conclusions have gradually been revealed. Here they are brought together and summarised.

Our perceptions of the context of systems development greatly influence how we view the process of developing systems. A perusal of the literature suggests that the metaphorical assumptions we make when interpreting an organisational context of information systems development can impact on the way we view the process of development. For example, if we view an organisation as a mechanistic entity, then we are likely to view the process of development as an engineering problem and a process of problem-solving. Simplistically, but not unreasonably, a link can be drawn between organic perceptions of the context and viewing change as growth or evolution, cultural metaphors of the organisation and a perception of change as learning, and a view of the organisation as a political entity and an assumption that change in organisations is a process of negotiation.

So, what contextual elements should be considered when creating an adequate process model? Specific relevant elements were identified from the literature and observations from the field and included: the purpose of the organisation and place in the wider context; the technical/process infrastructure; roles; authority relationships; and cultural structures. These elements reflect an emphasis on the political and cultural aspects of organisations, but a recognition of other aspects. The primary systems developers and users involved with the LSP, the OPC and CIPU were introduced as at the beginning of the project using this contextual model as a framework. Using the same framework to analyse both organisations highlighted the differences between them. Broadly, the OPC was a fairly bureaucratic professional and relatively stable organisation with little experience with computerised technologies. In the eyes of the CIPU, they could be described as a group “with strong traditions” and “afraid of change” associated with the implementation of the computerised technology involved with the LSP. On the other hand, the CIPU, while still largely a professional organisation, was technocratic rather than bureaucratic and dynamic (less stable) with strong technical experience and expertise. The OPC interpreted the constantly shifting roles, responsibilities and staff as reflecting a lack of commitment and perseverance and described the unit as “computer tech-heads”. The description of how the process of the LSP unfolded is basically a story of the interaction between two different units with differing frames of reference and these descriptions of the context sets the scene. Other players included executive management, Parliament and Cabinet, and other government agencies.

The content (output and outcomes) of the LSP was established via micro social processes which included the active (as well as passive) user involvement, the negotiation of meaning and interest, the creation and sustainment of coalitions of

meaning and interest and the application of hierarchical and expert authority. This analysis of the process stands in contrast to most literature on systems development, which ignores or relegates discussions of these elements to euphemisms such as “marketing”, “top management commitment” or “change management”. An examination of these social processes reveals that, at a micro level, the process was emergent. These observations build on those of Walsham and others who have focused on these social processes and emphasised their importance.

Forms of temporality	As related to the systems development process	Theoretical concepts employed to explain processes
Durée (micro social processes)	Interaction between those involved in systems development, including management, users, systems developers, contractors and consultants	Active and passive user participation; Negotiation of meaning and interest, Creation and sustainment of coalitions of meaning and interest, the application of hierarchical and expert authority
Dasien (lifecycle of organism)	The relationship between planning and implementation; a critical examination of the SDLC model and alternatives	Chronological and hierarchical emergence
Longue Durée (institutional time)	The creation and sustainment of negotiated social order and the broader processes of organisational change associated with technical developments	Structural frames, realms of action and movement between them; structuration theory

Table 5.1: The relationship between levels of analysis

However, change does not just occur at this micro level. Pettigrew (1985) suggests we should look at multiple levels for a contextualist analysis and the same is true of a “processual” analysis. Giddens (1979) suggested there are three forms of temporality, and this micro level analysis reflects only one of these levels. This dissertation has looked at the other two macro forms of temporality and the linkages between them and the micro processes, or *durée*. The *dasien* of systems development is generally described with the systems development lifecycle (SDLC) model and Part 4 illustrated that in most cases this model was inadequate. An adequate model of the *dasien* of systems development should incorporate hierarchical and chronological emergence and alternative models were assessed according to whether they did. Orlikowski (1996 and Orlikowski and Hofman 1997) and Barley (1997) focused on the *longue durée* and the development of social institutions, and this is the focus of Chapter 4.5. The concepts of hierarchical and chronological emergence are linked back to Giddens’ structuration theory through the concepts of structural frames and realms of action. This macro model can be viewed as another application of the

themes that have recently been articulated by Orlikowski and Barley in more detail and represent structuration in motion.

The process of systems development is usually represented at the *dasien* level of temporality, though there are connections between this macro process and the wider institutional level and the micro social processes. The relationship between these levels is illustrated in Table 5.1.

An important issue at the middle, *dasien* level is the relationship between planning and implementation. The importance of this issue becomes apparent when examining the term, *change management*. “Change management” is often used in relation to systems development projects such as the LSP, but an examination of the use of the term in both the literature and practice reveals confusion as to precisely what it means. Three distinct interpretations can be identified:

1. systems development as a process of managing change (ie, the management of projects *is* the management of change);
2. actions to ensure planned outcomes of projects are achieved (ie, change management is an activity *within* projects); and
3. actions to control and manage changes to the specifications of a systems development project that have already been established and agreed to (often also called configuration management or change control).

This confusion indicated problems in how we view the process of systems development. The commonly used SDLC model is used almost unquestioningly, despite its many identified problems. Those involved with the LSP were not uncommon in using the model and would have been unusual if they had not.

The irony is that the standard SDLC model does not even reflect the way that a project such as the LSP unfolds in reality, even if it is used. Westrup (1995) noted that systems developers often had more organisational understanding than their models imply and suggested this was a good thing. If a systems analyst produced a process diagram of the actual systems development process, it would look more like the emergent approaches identified in Chapters 4.3 and 4.4 than the SDLC model and reflect chronological and hierarchical emergence through the use of iterative, incremental and participative approaches. These approaches reflect the reality of actual systems development activities and, if used explicitly and creatively, could provide a much better alternative to obtaining the benefits from computerised technology than the large-scale project developments supported by the systems development lifecycle approach. By actively supporting the chronological and hierarchical emergence required, such alternative models would provide systems

developers with better support during the process of development as well as justifications for their actions.

Change management, as it is conceived *within* systems development projects, is indicative of this problematic conception of process. Using alternative ways of looking at the process, where the negotiated nature of social order is recognised or implicitly allowed for, the concept of change management as per the top-down approach is not an issue. These alternative approaches tend to emphasise the emergent, negotiated nature of change. If change is effectively negotiated, then a top-down approach to change management is meaningless. Changes are negotiated and established early in the project through the active involvement of all participants in formal forums. It does not need to be "communicated" and the importance of emphasising the corporate vision is reduced, as all who affect or who are affected by the changes had an active part in their creation. Thus, it is not surprising that texts which refer to different models of process tend not to mention the concept, and if they do, they refer to the emergent concept of change management.

How we conceive change management has significant implications for systems development (or visa versa). The top-down approach to change management insinuates that the content of change is relatively fixed and that change management activities focus on the implementation of these plans. It suffers from the same top-down rationalistic delusions of the systems development lifecycle model. The content of the changes are fixed, then implemented and those affected by the changes need to be brought into line with them. It does not recognise the emergent nature of planning and implementation nor the negotiated/socially dynamic nature of change and even tends to ignore the dynamic nature of context.

While the emergent approach to change management makes more sense generally, though, it does not make sense within the context of projects such as the LSP. This is because the emergent approach to change does not see major planned organisational changes as being feasible, especially within the scope of a single project such as the LSP. In other words, the LSP project would not have been initiated in its current form if the emergent approach to change management had been adopted. Instead, it probably would have formed a series of smaller scale projects. These smaller scale projects could have used any or all of the alternative approaches to systems development identified in Chapter 4.4. For example, an early JAD session may have identified the need to implement camera ready processes into the organisation and an improvement in the office's computing and network capacity. After these projects had been planned, further reviews may have identified other issues which could have promoted further developments, such as the creation of a database of legislation and ways of including amendments into these electronic versions. In this way, what is

quite a large, complex and therefore risky project is subdivided into many smaller and less risky projects. To a certain degree, this occurred through the identification of the Legislation Publication Project as a later and separate project, but it could have done to a much greater degree. The delays encountered by the LSP may have been avoided if more simple steps had been taken first and what was essentially an immature technology (textual databases and editors) was allowed to mature.

Of course, life is easier in hindsight, and these comments should not be taken as a criticism of the approaches adopted by the LSP team and others involved in the project, but a suggestion of alternative scenarios.

Combined with strong strategic goals, the emergent approach to change including a large number of small projects makes a viable alternative to the high risks associated with larger scale systems development projects such as the LSP. Such an approach would recognise the need to actively involve those affected by the change in the defining and creation stages of implementing change, as Mumford's (1995) ETHICS suggests. The process could involve the principles of RAD, through the recognition that each project is only one in a series, the extensive use of prototypes and the short project lifecycles. It may also include the recognition that the process involves a number of parallel tasks which are the responsibility of different groups, such as illustrated by Eason's (1988) multilinear model or Smyrk's (1997) output/outcome model. This stream of projects would be part of attempts to reach a longer term goal such as automatic consolidation within five years.

Throughout this series of projects, the members of the OPC would have had the chance to build up a more in-depth understanding of computing technology, so they could make better contributions to the later development. Instead of commenting that they found using the mouse intrusive when looking at SGML editors, their greater experience of general word processing technology would have given them a better basis on which to assess such editors and respond more constructively. There would have been less technical risk as the SGML technology would have been more mature by the time of investment and the long delays in the project and frustration caused by undelivered promises could have been avoided or at least minimised. Would the same or similar developments have been made without the establishment of such a large project as the LSP? Potentially, yes, if combined with an effective and ongoing change program process involving the right people. A series of RAD cycles could have formed part of this process.

Pettigrew (1985) advised that one should not focus on projects as the unit of change for change does not operate at this level. This research project illustrated the other side of the equation: that single projects cannot be associated with significant

organisational changes. These observations are hardly new but do not seem to have influenced the Tasmanian State Service. Significant organisational changes are best viewed as part of a series of projects, the dynamics of which cannot be determined entirely upfront. Projects using the systems development lifecycle may be used for the parts of this stream which are considered predictable, but in the main, other project process models, such as that underlying RAD, may be more appropriate. In this way, the change process could have reflected what Orlikowski and Hofman (1997) termed an improvisational model of development.

In other words, there needs to be a focus on the process of change and scoping projects within this process, rather than viewing the whole process as one project. Such a process could be termed *change program management* and would probably include elements of both project management and strategic planning. Smyrk's (1997) differentiation of a business initiative from an infrastructure project, the latter being a subset of the former, is a good start, but it only includes systems development sub-processes as one a range of parallel sub-processes, and does not examine how these sub-processes interact. There is a need to show how a series of projects can form a business initiative, rather than a one-off initiative planned in a top-down manner. This is what Orlikowski and Hofman (1997) referred to as an improvisational model of change management. They illustrated how this chronologically and hierarchically emergent process can unfold by describing a case study when such a process was employed. This research project complements their observations by showing both how such emergent processes unfold, with reference to the REMUS case study, and also the consequences of ignoring chronological and hierarchical emergence in both the HRMIS and LSP case studies. Orlikowski and Hofman's recent publication of their observations suggests the observations of the LSP, REMUS and HRMIS case studies have wider theoretical significance.

Like Orlikowski and Barley's recent work, this project has illustrated structuration in action. Particularly, it has illustrated that structuration occurs hierarchically as well as chronologically as individuals interpret contextual issues as an objective, given "fact" or within their realm of action. An individual's realm of action is bounded by a structural frame, which marks the boundary between their area of choice and areas in which they perceive they have no choice. In other words, embedded in the structural frame was the imperatives of the context which mediate choice and action. The boundary between someone's realm of action and the structural frame which surrounds it is negotiable, changes over time and differs according to one's social position and individual perceptions.

These concepts describe the *longue durée*, or institutional time as social networks change over time. It illustrates how broader contextual changes can influence internal

organisational changes and how authority and previous actions can structure the behaviour of others. It describes the processes of objectification as social orders are created and sustained, so adding to the observations of those employing Latour's Actor Network theory and providing an alternative to the commonly employed "unfreeze-change-refreeze" model of organisational change.

These multiple levels of analysis of systems development give a deep insight into the processes of technologically induced change in organisations today. It conceptually describes the process by which government information systems are planned, developed and implemented. This description stands in contrast to the way it is often depicted and suggests some ways of improving the management of this process.

5.2 Specific Contributions

Specific contributions of this dissertation include:

- a description of legislative drafting during a time of change. Surprisingly, given the critical nature of legislative drafting in public administration and the workings of government, there is very little written about the role of drafters and their work processes;
- a focus on two large multi-department information systems. Given current moves towards whole-of-government systems, multi-department systems are likely to be more common. However, to date they have been relatively rare, largely due to the complexities and difficulties in coordinating activities in multiple agencies. The focus here has been on the process by which such systems are planned, developed and implemented and this will have significance for current and future multi-department information systems initiatives;
- a detailed examination of the micro-social processes involved in the LSP. These processes include the negotiation of meaning and interest, the creation and sustainment of coalitions of meaning and interest, active and passive user involvement and the application of expertise and hierarchical authority. Through an examination of the literature on systems development, it is illustrated that these micro-processes seem to be common to all such information systems projects;
- an examination of the linkage between these micro-social processes and the macro-processes of change which underlie systems development projects using Giddens' (1973) concepts of *durée*, *dasien* and *longue durée* as a framework. Most literature in this area focuses on only one level, while there is a focus on multiple levels and linkages are made between them.

- the application of Giddens' structuration theory to information systems development. Although not a novel contribution in itself, it adds to the growing body of literature illustrating the utility of the concepts involved. This application has involved a description of how the process of structuration unfolds in relation to an in-depth case study through the concepts of *realms of action* and *structural frames*. These concepts illustrate the manner in which new structures are created through social interaction over time, as illustrated in Chapter 4.5.
- the differentiation between chronological and hierarchical emergence.
- a critical examination of the commonly used macro-level process model underlying systems development initiatives, the systems development lifecycle, and an identification of alternative models of process which could be used instead of the systems development lifecycle.

5.3 Implications for theory

This dissertation has introduced a descriptive process of the systems development process. This echoes work by Barley and Tolbert (1997) and Orlikowski and Hofman (1997), and illustrates structuration in action and hierarchical and chronological emergence. Orlikowski and Hofman described two approaches to change: planned and improvisational, and illustrated an improvisational model of change in relation to a case study of a large consulting firm. In contrast, the LSP provides an example of the former approach to change and illustrates the risks of doing so. Juxtaposed with this, the REMUS case study outlines the implementation of a large, multi-government department human resource management system and provides another example of the alternative approach. Although there was a goal of improved personnel and payroll facilities to direct actions, there were often not firm plans and changes were largely directed by those who would implement and utilise them. Significantly, many of those who were influential in the REMUS project were relatively unexposed to recognised standards of information systems best practice and this may have been a good thing.

Quite simply, some of the standard and well-recognised models underlying systems development and deployment do not provide an adequate guide for action. Ironically, those involved in the LSP did try and support active user participation, iteration and incremental development, but these activities were limited by the standard tools and techniques they deployed to assist them. These tools and techniques did not provide the adequate means to articulate and justify many essential or important actions. As Westrup (1995) commented, many systems developers have better organisational awareness than their tools imply. If this is the case, some of the dominant tools could

actually be interfering with the application of this awareness and alternative models should be investigated. Possible alternatives might include rapid application development, with its emphasis on short project cycle times, prototyping and joint application design sessions, or STEPS, an elaboration of the ETHICS methodology, which promotes not only active user participation but also incremental and iterative development.

If we accept that systems development is, at a micro level, a process of social interaction as people negotiate interest and meaning and apply expertise and authority, there are significant implications for viewing the overall process. If systems development involves such negotiation, projects are unlikely to be defined completely upfront as they are subject to ongoing sensemaking and negotiation. Plans are likely to be vague in order to create coalitions of interest and ongoing negotiation determines what these plans entail in practice.

If the positive potential of advanced technology is to be realized, technocratic organisations must be democratized and technocratic rationality must be replaced by a more substantive political rationality that considers long-term implications and much broader social and political parameters. In this long and difficult process of societal restructuring, there can be no substitute for the politics of coalition building, debate, and negotiation among various constituencies... (Burris 1993: p 180)

The context models introduced in Part 2 identified the interaction of a systems development project such as the LSP with its organisational context but they do not illustrate how this interaction occurs over time. The process model should convey the idea of movement over time and, while the context models strongly insinuate such dynamics, they are essentially snapshot portraits. The concepts raised in Part 3 – the legitimization of people's actions through the negotiation of interest and meaning, the imparting of expertise and application of authority – show how these dynamics occur at the micro level of day-to-day social interaction. Descriptions of the LSP have illustrated how these concepts explain how the content of a change initiative such as the LSP emerge via these processes.

The processes occur throughout projects such as the LSP. Even after plans are made defining such projects, aspects of it, or even the whole plan, may still be up for negotiation or subjected to changes by those with authority or perceived expertise. As coalitions of interest and meaning need to be established and it is difficult to fully articulate precise plans anyway, changes may also occur simply through the process of interpreting what these plans are. Thus the goals of the project can be redefined.

The rationalistic approach that dominates the systems development literature ignores this. As discussed in detail in Part 4, plans are assumed to be defined and then implemented. Such rationalistic assumptions also forget that the definition and implementation of such projects is achieved through social interactions and almost completely ignore the processes of legitimization through negotiation, authority and

expertise. They relegate problems to issues of "resistance", "a lack of top management commitment", and "a lack of training", so painting over the problems and hiding the fundamental issues. "It's not the problem of the plan that it cannot be implemented", they are arguing. "People just need to understand and be committed to the path outlined by it and not resist it and change management is the solution." The answers to many problems in systems development are simplistic and reflect a lack of recognition of the social processes that underlie systems development. It is not the decor that needs adjusting but the house of managerial rationalism in relation to systems development needs to be reviewed and rebuilt if the storms of low project success rates are to be weathered.

However, while highlighting the importance of the context of change, such emergent approaches ignore one important aspect of the context – the context of the managers and systems developers. Their context strongly promotes a top-down, rationalistic approach to systems development. Despite the problems of this approach, auditing procedures of their work and the pressures of the tendering process reinforces the top-down approach. Their work is assessed according to rationalistic assumptions. Systems developers do not just use methodologies because they help them guide their tasks- they use them to illustrate that they have followed a standard procedure and retrospectively fit their tasks to such methodologies for auditing purposes to illustrate standard accepted procedures have been followed. Rationalistic external reviews relying on top-down assumptions (eg judging the implementation of policies according to how well they meet the preset goals rather than present needs) promotes the continuation of such top-down approaches. Operating in a large, bureaucratic organisation, there are strong pressures to employ top-down approaches to systems development. Nevertheless, if they can justify their actions, as those involved in the CHIPs project in British Columbia did, alternative approaches may prove fruitful. This issue illustrates that one cannot separate theory from practice.

5.4 Implications for practice

In April 1997, as I was buried deep in the early drafts of this dissertation, the Premier of Tasmania announced the initiation of broad ranging initiatives involving information technology and its application. This so-called *Directions Statement* incorporated and initiated a number of significant projects, such as the development of a one-stop government service delivery unit, the diffusion of computers into schools and new telecommunications infrastructure.

With experience in managing projects, the CIPU found themselves thrust to the centre of the initiatives. Without the resources to manage such a large number of projects themselves, the CIPU took on a new role of providing advice to those directly

responsible for those projects. Along with an information policy and information resource management support unit, the Information Strategy Unit (ISU), they developed project management guidelines based on their experiences with the LSP and other IT development projects.

In July 1997, I was asked to join the ISU and one of my tasks was to help with the development and dissemination of these guidelines. By this stage, parts of this dissertation had been presented to various people involved in the LSP for comment and review but the broad conclusions had only been mentioned in passing when specific issues arose. Employed by the ISU, I was suddenly expected to become an active participant in a situation where I had previously been largely a passive observer.

In my new role, I was involved in developing the guidelines for use by people involved with the large Directions projects. In many ways, the similarities between the early days (daze?) of the LSP and these new large projects were frightening. Many of them were very broad initiatives from the top of the organisation and were being tackled with the one-shot, monolithic approach favoured by the CIPU's formal project management and systems development approaches generally. Pressed to take on a more active role, at the same time I was discussing the conclusions of this research project, I suggested that these new projects needed to incorporate far more chronological and hierarchical emergence (though not in so many words). In particular, this entailed more iteration and an incremental approach rather than the usual monolithic approach to managing change.

When those involved in information systems are approached with a vision for change, they tend to initiate a project to enable them and view associated organisational changes within the bounds of this initiative. That is, they define a desired technical and organisational changes within the scope of a project. Yet these standard methods of managing projects simply do not cope with the effective facilitation of significant organisational changes. They view change as predictable and plannable while organisational change tends to be emergent. In adopting the SDLC process model of technological development projects, we ignore the socially negotiated nature of the process and assume that changes can be planned almost entirely before they are enacted. For example, we assume we can determine the requirements of those who will be affected by the changes at the beginning of the process but this is often unrealistic. In response to these concerns, I made the following recommendations:

5.4.1 Coping with emergent and Unanticipated Change

“Change is inevitable - except from vending machines.” No matter how well a project is planned, there are likely to be unforeseen circumstances or issues that simply cannot be determined up-front. We can divide types of change into two major categories - planned and unplanned. We can further subdivide unplanned change into emergent and unanticipated based primarily on our awareness and control of the changing circumstances:

- **Planned.** (Change that is planned and basically implemented as anticipated);
- **Emergent.** (A proactive response to unforeseen circumstances. For example, additional or conflicting requirements may become apparent and are responded to. Alternatively, circumstances may change);
- **Unanticipated.** (That is, change that is unplanned and unforeseen. For example, people may use implemented technology in a way that was not intended)²⁸.

Unplanned change is likely to happen, no matter how competent and prepared project managers are. Governments change or are restructured. Technology develops or becomes redundant. People’s opinions or viewpoints change. Changes that involve negotiation or substantial learning (either organisationally or individually) tend to involve a great deal of emergent or unanticipated change. The outcomes of learning or negotiation can be anticipated but not wholly planned, as they tend to emerge over time.

Signs of a need to carefully consider the management of emergent or unanticipated issues include:

- difficulties in determining project requirements in depth;
- the facilitation or acceptance of change by those affected by it is seen to be a major issue (so indicating a need for major negotiation and/or learning);
- a high degree of technical or other types of innovation;
- a rapidly changing or vague project context.

Unplanned change does not have to be unmanaged. Emergent and unanticipated issues can be addressed either within the scope of a single project or by translating a major initiative for change (a vision for change) into a number of interlinked projects, rather than one monolithic one.

In practice, dealing with such issues within the scope of a project involves:

- anticipating and planning for possible changes through risk analysis contingency plans;
- keeping track of emerging or unanticipated issues through issues management procedures;
- bringing issues which could have a major impact on the nature or substance of the project to the project steering committee so they can re-evaluate the project or make adjustments;
- using an iterative process of change within the scope of a single project. An example of such an approach for information systems development projects is Rapid Application Development (RAD). RAD is highly recommended by international consulting groups such as Metagroup for projects involving innovation or organisational changes, such as datawarehousing.

Alternatively, translating a major initiative for change (a vision for change) into a number of interlinked projects, rather than one monolithic one, can help avoid emergent or unanticipated issues. In many cases “prevention is better than a cure” and if a proposed project is likely to involve substantial emergent or unanticipated issues, the proposed scope of the project should be carefully re-evaluated and possibly broken into a number of smaller projects.

That is, visions for change are translated into a series of project cycles rather than one project cycle. Breaking up change initiatives into “bite sized pieces” ensures project managers are not given more than they can chew with the tools they have. Larger projects are exponentially more risky than smaller ones as they involve far more complexity.

²⁸ Wanda Orlikowski and Debra Hofman 1997

In practice, this simply involves recognition of and planning for desired outcomes on a large-scale, strategic level without committing to a particular set of implementation tactics (including the number, nature or scope of projects down the track).

A series of interlinked projects is less risky than one larger one for several reasons. Firstly, complexity is reduced by dividing the change initiatives into smaller areas of action. Fewer people and tasks are involved. Secondly, it is easier to produce identifiable outputs and outcomes from smaller projects, which can be used to feed into later projects. Thus, even if the full objectives of the change initiative are not met, identifiable achievements are. Thirdly, it can be easier to respond to changing or unanticipated circumstances, as projects lifecycles are much shorter and new or emerging issues can be pursued through the planning stages of future projects. Fourthly, it allows for negotiation or substantial learning. These are both integral to many change initiatives but are not always well supported.

One possible risk of this approach is that those involved with the series of projects may lose sight of the broader objectives of the change or simply not achieve them. Firstly, it should be pointed out that large projects both in this government and in others have a very poor record of success and commonly do not achieve their intended objectives²⁹.

This risk can be mitigated by carefully coordinating the series of projects, either by linking them by an overarching project, or carefully coordinating them with strategic planning processes.

Related projects may be coordinated by relegating them to the status of sub-projects in a larger project. This is suitable when the objectives and tasks involved with each sub-project are relatively understood, but is less suitable with projects involving substantial innovation, negotiation or complex issues which are not greatly understood.

Alternatively, the projects may be viewed as products of a continued process of strategic planning which is recognised to be an emergent process³⁰. This approach is more suitable for projects involving innovation, negotiation and complexity that cannot be adequately anticipated up-front. The strategic planning process should include those involved with the projects and should be a carefully managed and ongoing process which carefully reviews past progress and well as future directions. If strategic planning is viewed as a one-off or periodic exercise for top managers, or focuses only on longer term time horizons, there can be little relationship between strategic planning and project management processes.

Sometimes major change initiatives are translated into single projects. Project managers should be aware that this approach is likely to involve substantial problems and are extremely unlikely to be delivered on time and within budget. The approaches outlined above can help mitigate these risks to a degree.

The latter approach, focusing on the close relationship between strategic planning processes and projects, can result in the more effective implementation of planned change initiatives. However, strategic planning processes are outside the scope of project management. If these processes are non-existent or not effectively in place, those involved in planning the change initiatives might find it easier to obtain commitment (ie funding and resources) if they can define set deliverables, timeframes and activities. In this case, the former tactics would be more appropriate for managing emergent or unanticipated issues. As with many project management decisions, an adequate appreciation of the project context is crucial.

This section was incorporated into the Service-wide Project Management Guidelines v3, released in November 1997. These guidelines were extensively reviewed by a range of project managers and have been widely adopted. Such acceptance of the practical conclusions of this research validates their relevance. Upon reflections, changes to this section would include more detailed suggestions as to how a process of change program management would operate so that a related projects could be

²⁹ see Nada Korac-Boisvert and Alex Kouzmin (1995)

³⁰ see Henry Mintzberg (1993)

effectively combined. This change program management would include elements of strategic planning and project management and would operate at the level of project steering committees or the project leader of large projects. This remains an area for further research.

5.5 Future Research

Other areas for possible future research include:

- a follow-up study of the OPC examining how they utilise the technology over time and any associated changes. Such a study would directly echo work by Orlikowski (1996) and Orlikowski and Hofman (1997) in a differing organisational context, so aiding the generalisability of their concepts;
- An examination of the literature has revealed very little consideration on how the use of wordprocessing technologies could be changing the creative process of writing and writing styles. The diffusion of the printing press is said to have led to changes in writing styles, as the resulting text was intended to be read visually, rather than orally but it was difficult to find anything examining such issues in relation to the diffusion of computerised technologies. Also related to this is the use that people make of computerised technology in the process of writing, the conceptual approach they take and requirements they have. This topic was an area of relevance for the drafters and of interest to anyone using computerised technology in the writing process.
- An examination of the implications of the EnAct system and computerisation generally on the broader processes of enactment and parliamentary processes.
- Further analysis of the use of the systems development lifecycle and alternative process models in practice;

Such future research could add to the growing and increasingly relevant stream of literature focusing on the process of implementing technologically induced change in organisations and the application of computerised technology in an organisational context.

5.6 Concluding reflections

In essence, while the technology and its application through information systems has become increasingly sophisticated, the processes we use to manage its development and implementation are lagging behind and are contributing to problems in systems development. Those directly involved in the LSP tried to ensure they were applying standards of best practice in this area. They utilised reputable consultants as advisers,

sought, gained and maintained quality certification and used well-recognised methodologies. From where I stood, the personal competence of the individuals involved was not an issue. Yet the project has suffered significant delays and faced real user acceptance risks. A surface glance would suggest these problems are attributable to the use of novel technology and a lack of user participation, yet these observations, while not totally wrong, are simplistic and do not provide us with lessons which could be applied to other situations. The problems can be linked to how we view the process of change and how we define an information systems development project.

In essence, those involved in systems development are not only applying what Kling (1992) would term a discrete-entity model of process, but enacting it. This dissertation has provided an alternative descriptive model which incorporates multiple levels of analysis and a critical examination of existing models and alternatives. The main conclusions contribute to a growing stream of literature challenging the existing dominant approach by further articulating the issues involved and linking them to in-depth observations of systems development initiatives.

Appendix 1

Review of commonly available systems development literature (published since 1990)

author	title	year	SDLC	alternatives identified	A	B	C
Avison, D & Fitzgerald, G	Information Systems Development: Methodologies, Techniques and tools	1995	1	y - within discussion of methodologies	1	1	1
Burch, JG	Systems Analysis, Design and Implementation	1992	1	y - prototyping	1	0	0
Crowe, M, Deeby, R et al	Constructing Systems and Information	1997	0	y - "evolutionary and process-based approach"	1	1	1
DeWitz	Systems Analysis and design and the transition to objects	1996	0	y - OO + RAD	1	1	1
Eliason, AL	Systems Development: Analysis, Design and Implementation	1990	1	y - prototyping	1	0	0.5
Gibson, R	Managing Computer Projects: Avoiding the pitfalls	1992	1	n	0	0	0
Harris, D	Systems analysis and design: a project approach	1995	1	n	0	0	0.5
Hawryzkiewicz, I	Introduction to Systems Analysis and Design	1994	1	y - p'type evol'ut'y design, staged design, team dev't	1	1	1
Hoffer, JA, George JF et al	Modern Systems Analysis and Design	1996	1	y - prototyping, OO, RAD	1	0	0.5
Immon, WH & Caplan, JH	Information Systems Architecture: Development in the 90s	1993	1	n	0	0	0
Jorden, EW & Machesky, JJ	Systems Development	1990	1	n	0	0	0.5
Kendall, KE & Kendall, JE	Systems Analysis and Design	1995	1	n - prototyping, ETHICS & SSM mentioned	1	0	0
Laudon, KC & Laudon, LP	Analysis and Design of Business Information Systems	1995	1	y - prototyping, use of application packages, EUC	1	1	0.5
Martin, MP	Analysis and Design of Business Information Systems	1995	1	y - prototyping	1	0	0.5
Mason, D & Willcocks, L	Systems analysis, systems design	1994	1	y	1	0	0
McLeod, G & Smith, D	Managing Information Technology Projects	1996	1	n - incorporated into lifecycle approach	1	0	0.5
Moynihn, E	Business Management and Systems Analysis	1993	1	y - prototyping	0	0	0
Mumford, E	Effective Systems Design and Requirements Analysis	1995	0	y - participative approach	0	0	1
Olle, WT, Hagelstein J et al	Information Systems Methodologies: A Framework for Understanding	1991	1	y	1	1	0
Powers, MJ	Structured Systems Development: analysis, design, implementation	1991	1	y	1	1	0
Robb, AF	The Mgt Guide to the Selection and Implementation of Computer Systems	1992	1	n	0	0	0
Shelly, GB, Cashman, TJ et al	Systems analysis and design	1995	1	no	0	0	0.5
Taylor, DA	Object-oriented information systems: planning and design	1992	1	y - prototyping	1	0	0.5
Walsham, G	Interpreting Information Systems in Organisations	1993	0	y - ETHICS, SSM and "Scand'n approach"	0	1	1
	1=YES; 0= NO; 0.5 = NOMINAL	24	20	total yes: 15 prototyping: 9	15	8	11
A: iteration=prototyping + iteration							
B: incrementalism = changing requirements expected; ongoing planning, not just top-down; not strict separation between planning & action; modular development and/or incremental development/ implementation strategy							
C: participation = active user involvement/power throughout process; recognition of social processes; systems developers facilitators not experts							

Appendix 2

OPC Processes at the beginning and end of the LSP

(source: LSP Impact Analysis Report)

Major issues identified include:

- For the OPC to obtain maximum benefit from the system, there needs to be a commitment by the drafters to use the system. *How will this be ensured?*
- There are only a few written procedures and policies relating to the current business processes within the OPC. *Who will be responsible for the preparation of procedural documentation - including its ongoing management?*
- Workflow tools provided with EnAct structure the tasks associated with drafting into workflow steps. *Who will be responsible for performing these steps for each drafting file?*
- It is clearly stated in the LSP Business Case that an objective of the project is that the presentation of amending legislation to Parliament will be improved. *Will any of the products of the system be used to present to Parliament? What procedures will be established to facilitate this?*
- As new tasks within the OPC are identified, they will need to be assigned to an OPC staff member(s). The OPC position descriptions will need to be reviewed in the light of new tasks that are introduced and old tasks that have become obsolete. *Which position descriptions need updating? Do any positions need re-classifying? Is there a need to establish any new positions? Can a career path be created from these changes?*
- EnAct will introduce a new way to produce and manage legislation. The OPC staff will experience a “familiarisation” period as they adapt to the new processes. This may decrease the throughput of the OPC for a while. *How will the CPC inform all affected parties of changes to any current OPC procedures? How will affected parties be notified that there may be an initial delay in turn around time of legislation while the OPC becomes familiar with the new system?*
- All legislation, including Private Members Bills which have been drafted externally, will need to conform to the prescribed structure/format to enable the legislation to be saved as SGML. *How will the CPC ensure the format and structure requirements of EnAct are satisfied for legislation which is drafted externally? How will these drafting standards be communicated to external drafters?*
- Future changes to OPC procedures will require an assessment to be made of the impact, if any, in relation to EnAct. If there is a need for a requested change to the system the OPC will need to determine how they will fund the change. *How will changes requested from within the OPC be funded? How will changes requested from outside of the OPC (eg. Ministerial direction) be funded?* (Impact Analysis Report v 1.0 5 September 1997).

	CURRENT	FUTURE	ANALYSIS
Systems	<p>All staff within the OPC have access to a personal computer and they are currently used in the following manner:</p> <ul style="list-style-type: none"> • Word 6 is sometimes used to draft legislation in a basic word processing format; • Word 6 templates and styles are used to prepare the drafted legislation according to a Camera Ready specification; and • Excel and Word 6 are used to maintain indexes of information relating to legislation drafted and processed by the OPC. 	<p>EnAct ...will provide:</p> <ul style="list-style-type: none"> • drafting tools; • workflow tools to monitor the progress of each legislative drafting task; • management of documents relating to each legislative drafting task; • online access to legislation for research/reference purposes; • the ability for drafters to maintain full control of the workflow for each legislative drafting task; the ability for drafters to maintain complete responsibility for document management; and • reporting facilities for the CPC to obtain information in relation to drafting files allocated within the office. 	<p>Note this analysis of the current processes occurred immediately prior to the implementation of the EnAct system. At the beginning of the LSP, only administrative assistants used computers on a regular basis, and generally they only used them for word processing facilities.</p>
Issues - Current Processes General	<p>All legislation (with the exception of some Private Members Bills) is drafted by the OPC. After receiving instructions the drafting work is allocated to one of 7 drafters. Upon completing initial research and consultation with an Instructing Officer (as appropriate) draft legislation is prepared. Generally the drafts are hand written with occasional reference to standard paper templates/proformas, dictated using a dictaphone, or the drafts are word processed in a basic format. The draft is then prepared according to a Camera Ready specification. This electronic copy is proof read against the drafter's original typed/hand written version and an editorial check is performed, if required.</p> <p>Once a version of the legislation has been agreed to by the Instructing Officer the drafter assumes responsibility for the accurate completion of the draft legislation and for any further revisions that are requested to be made to the legislation. However, the drafter does not take an active role in finalisation for Camera Ready printing. These tasks are completed by the Executive Officer, Records Clerk, Administrative Assistants and Proofreaders.</p>	<p>The CPC will allocate "drafting files". Drafters will only have access to drafting files that have been allocated to them. The CPC can review resource allocation within the OPC and monitor the progress of all legislative tasks.</p> <p>The system allows drafters to directly input legislation into electronic form with purpose built drafting tools, which incorporate integrity checks (ie structure / referencing / numbering). Drafters can enter validated references while drafting or delay validation to a later time.</p> <p>Each drafting file contains:</p> <ul style="list-style-type: none"> • relevant documents (eg. drafts, letters, memos) • workflow information (eg. approval dates, relevant Minister) 	<p>Note the system focus of this analysis. The purpose of this analysis was to highlight what procedural changes could occur as a result of the EnAct system. The non technical elements of these changes are still being determined. Other issues suggested include:</p> <ul style="list-style-type: none"> • What will be the role of proofreaders
Bills	<p>Parliamentary Bills are all drafted by the OPC. These Bills can originate from either house of Parliament. Prior to the drafted legislation being submitted to Parliament, a Bill number is allocated by the OPC. Any amendments to introduced Bills require the drafting of Proposed Amendments. Not all of these amendments are drafted by the OPC, as any Minister can introduce amendments to Bills under consideration. Any</p>	<p>EnAct maintains a Bill number index which can be accessed by drafters.</p> <p>The system allows for Proposed Amendments to be drafted externally to the system. These documents, which may be hand written, email-ed, or word processed, can be attached to the "drafting files". However, for smooth incorporation into the drafting file, the appropriate Word 6 templates must be</p>	<ul style="list-style-type: none"> • The assignment of Bill numbers, like many office administration tasks, has been the responsibility of the

	<p>amendments not drafted by the OPC are incorporated into the Bill as described below.</p> <p>Any Proposed Amendment agreed to by the originating house will be incorporated into the Bill. Any Proposed Amendment agreed to by the non-originating house will be used for the preparation of buff/green pages for presentation back to the originating house.</p> <p>As the Bill is passed by each house of Parliament a hard copy of the Bill is returned to the OPC. OPC prepares the relevant Camera Ready version, including appropriate certificates, ready for introduction into the next house.</p> <p>When both houses pass a Bill, an Act number is allocated by the Clerk of the Legislative Council. The OPC prepares a Camera Ready version of the Act known as the Vellum. The Vellum is proof read and reviewed prior to delivery to the Clerk of the Legislative Council.</p>	used.	<p>office's Records Clerk. What will his role and tasks now be?</p> <ul style="list-style-type: none"> • How will the green and buff coloured parliamentary amendments be managed in relation to the system? • How will proposed amendments that are emailed to the OPC be handled?
Amending Bills	<p>Most Amending Bills are drafted by the OPC.</p> <p>Drafters make amendments to the legislation by generally hand writing the changes. These are converted into an electronic form using Word 6 with specially written templates and styles to process the amendment drafts into the prescribed format.</p> <p>Once a version has been agreed with the Instructing Officer the drafter assumes responsibility for the accurate completion of the drafting process, but does not take an active role in its completion. These tasks are completed by the Executive Officer, Records Clerk, Administrative Assistants and Proof Readers.</p>	<p>Amending legislation is performed by using the drafting tools within the legislation system to edit or "mark up" the principal legislation which is to be amended. This involves: obtaining a copy of the Principal Legislation to be amended, consolidated to a selected point in time (known as "checking out" legislation); "locking" the Principal Legislation using a meaningful free-form message, so that other drafters attempting to amend the same legislation are notified that whole/parts of the legislation are currently in the process of being amended; creating a document which contains specific information relating to the Amending Bill, such as the long title, short title, and commencement and other substantive provisions. This document is referred to as the "stub" document; marking up and drafting any required amendments in the copy; converting the marked up copy into SGML which will create the Change Description Document (CDD); merging the "stub" with the relevant Change Description Documents (CDD's); and managing/modifying the CDD's.</p> <p>If there are changes to more than one piece of Principal Legislation then multiple Change Description Documents (CDD's) will be created.</p> <p>For discussion purposes, the Instructing Officer can be provided with a copy of the Principal Legislation as it will appear after the amendments have taken effect or in marked up form.</p>	<ul style="list-style-type: none"> • The previous procedures for creating paste-ups will no longer be required, though current versions of legislation will be. This entails new processes, as now electronic "paste-ups" are printed out from the EnAct database when changes are made. • Who will manage workflow tasks? This is still being resolved
Private Members' Bills	<p>Not all Private Members Bills are drafted by the OPC. If a Private Member does request the OPC to draft the Bill then the drafting work is allocated and processed in the same manner as principle legislation.</p> <p>If the Private Member elects not to use the OPC to initially draft the Bill,</p>	As per general	<p>Issues arising noted by the LSP team include:</p> <ul style="list-style-type: none"> * Who assigns bill numbers and how?

	the Clerk of the House of Assembly contacts the Records Clerk within the OPC to obtain a Bill number. If the legislation is passed by the originating house of Parliament, then the legislation needs to be reviewed by the OPC before it can be introduced in to the next house of Parliament. The drafters will generally make any changes to the agreed legislation so that it conforms to the current drafting standards adopted by the OPC.		• How will the OPC ensure the format and structural requirements of EnAct are satisfied for legislation which is drafted externally?
Statutory Rules	All Statutory Rules are drafted by the OPC as per usual, with the following minor variations: a Statutory Rule number is assigned; there is no requirement for Proposed Amendments and buff/green pages; Camera Ready versions are produced with provision for signatures of relevant bodies; Advice of Statutory Rule may be provided to the Executive Council Executive Council Minute may be prepared for the Executive Council; final Camera Ready copy is delivered to Printing Authority of Tasmania; and a Gazette Notice is prepared which requires authorisation.	EnAct maintains a Statutory Rule number index which can be accessed by drafters.	EnAct assumes that the drafter performs all of the Camera ready steps. The LSP team raised this issue and asked how this would be handled. This indicates an assumption that the system will dictate the working practices of the OPC.
Statute Law Revision Bills	All Statute Law Revision Bills are drafted by the OPC. These amendments have to pass through the Parliamentary process. A drafter may initiate a Statute Law Revision Bill, or the Records Clerk will indicate to the CPC when there are a reasonable number of amendments justifying the preparation of a Statute Law Revision Bill. Statute Law Revision Bills involve making amendments or repealing multiple pieces of principal legislation. Preparation of Statute Law Revision Bills is similar to Statutory Rules. However, there is no consultation with an Instructing Officer and the final version of any amended legislation may be circulated for comment to any affected Agencies.	EnAct will provide for Statute Law Revision Bills in a similar manner to Amending Bills. The time taken to prepare a Statute Law Revision Bill is not likely to reduce. Each piece of legislation to be amended needs to be processed by the same procedure as for an Amending Bill. Each of these individual amendments are then merged together using the CDD Manager to create one large CDD containing all of the required amendments. The more Acts that are amended, the more complex the process becomes. To offset this is the expectancy that the number of changes that fall into this category should reduce over time. Once all of the Tasmanian legislation is on the EnAct database the accuracy of the legislation will improve. All newly drafted legislation will need to conform to the defined required structure and format for legislation in EnAct, and all new cross references made to other Tasmanian legislation will be automatically checked.	• It may be more efficient to draft smaller Statute Law Revision Bills more frequently. How will changes to legislation not on the database be managed?
Changes under Leg'n Public'n Act 1996	All changes under the Acts Reprinting Act 1979 are prepared by the OPC. Under the Act these changes are not required to pass through the Parliamentary process. These changes are normally picked up when preparing a reprint or when drafters are doing their research and include changes such as: obsolete references; incorrect spelling; inconsistent punctuation; definitions out of order; and incorrect numbering. The updated copies of the legislation are forwarded to the Printing	At some stage (as determined by the CPC) the Legislation Publication Act 1996 will be commenced, which will repeal the Acts Reprinting Act 1979. EnAct will be used to prepare changes under the Legislation Publication Act 1996 as it would for Statute Law Revision Bills with the following exceptions: no "stub" document is produced; and no Amending Bill is produced Only a Change Description Document (CDD) is produced.	

	Authority of Tasmania.	With the ability to automatic consolidate legislation within EnAct the requirements for reprints will change.	
Drafting Standards	<p>Standards for the drafting and preparation of legislation include:</p> <ul style="list-style-type: none"> • previous legislation (precedents); • internally documented standards (limited); • standard paper templates/proformas; • individual drafting style of each drafter; • Camera Ready templates (Word 6); • Cabinet direction; and • Ministerial direction. 	<p>EnAct provides a series of steps which represents the logical work flow for the preparation and processing of legislation. A step may require the entry of one piece of information (such as a date); the entry of many pieces of information in a dialogue box or the creation of a Word document which contains the text of the legislation. In as many places as possible, a drop-down list of alternatives is supplied.</p> <p>The entering of the legislative text can be performed in a creative "free form" manner, or by adding each piece of the legislation in its final format eg. as headnotes, sections, paragraphs etc. Regardless of the method used to draft the legislation, it needs to be modified into the correct structure before it can be converted into SGML. This is the format required for all of the information in the legislation database.</p> <p>Although each step in the "workflow" is required, they will be completed over a period of time.</p>	<p>The EnAct system embodies a sophisticated level of drafting standards, and if these standards are to be changed, the system will have to also be changed. Such changes may be minor (eg changes to expiry clauses) or major (eg if the Legislative Council was abolished). Will these standards be documented? If so, how?</p>
Drafting Research	<p>Research tools used by drafters comprise, but are not restricted to: Bills currently under consideration by Parliament; "paste-ups" of current legislation; annual volumes; copies of reprints of legislation; copies of Statutory Rules; copies of legislation from other jurisdictions; legal sites on the World Wide Web (WWW); dictionary, thesaurus; Case law reports; discussion with other drafters within OPC; Parliamentary Committee reports; and other text books in the OPC library.</p>	<p>EnAct will allow the OPC drafting staff to access a copy of consolidated legislation at a specified date ie. current date, a date in the future or a date in the past. Access to copies of legislation from other states or from the WWW will remain unchanged. EnAct provides a tool with search facilities to allow drafters to look for: key words in legislation; styles in legislation; and subordinate legislation under Acts.</p> <p>When the publication of the Tasmanian consolidated legislation is available via the WWW, it is likely to provide more advanced search facilities that could be used by the drafters for research purposes.</p>	
Interaction with Instructing Officer	<p>Upon being allocated a drafting task the drafter will complete initial research and consultation with the Instructing Officer. This may involve several discussions. These may be conducted within the OPC, at the premises of the Instructing Officer, by phone, by correspondence or by email. The method and frequency of interaction with the Instructing Officer varies.</p> <p>Drafters with the OPC have individual methods of approach to liaising with each Instructing Officer. The level of interaction may also vary depending on the complexity of the legislation to be drafted, and the legal knowledge of the Instructing Officer.</p>	<p>The OPC drafting staff will still be able to continue these consultations when EnAct is implemented. It will not directly alter the current relationship between the drafters and the Instructing Officers. However, the system will provide several features which could be used by the drafters when interacting with the Instructing Officers, including: the ability to enter legislative text with the Instructing Officer present; the ability to "mark up" Principal Legislation with the Instructing Officer present ie. the changes to the legislation do not have to be described in plain English; provide the Instructing Officer with a copy of the final draft legislation in the correct style/format; and in the case of Amending Bills,</p>	<p>Will any products from the system (eg "marked up" versions) be used when drafters interact with instructing agencies or will such products be used to present to Parliament?</p>

		<p>provide the Instructing Officer with a copy of the Principal Act as it will appear after the amendments have taken effect, or a "marked up" copy of the Principal Act.</p> <p>It is possible that these features could be used in such a way to improve relationships between the OPC and Instructing Officers. It may also improve the drafting throughput of the OPC.</p>	
Commencement of Legislat'n	<p>Legislation can be commenced in several ways. All legislation contains a commencement provision which contains the wording that implies when whole/parts of the legislation commences.</p> <p>Acts may be commenced by: Royal Assent; On a day to be Proclaimed; On a day or days to be Proclaimed; On a Fixed date; On a Fixed Retrospective date; On a date if Royal Assent not granted by date; Dependent on Commencement of another Act; and Other, as appropriate.</p> <p>Statutory Rules may be commenced by: Notification in the Gazette; On a Fixed date; On a Fixed retrospective date (limited cases only); Dependent on Commencement of another Act/Statutory Rule; Notification under an Act; and Other, as appropriate.</p> <p>Currently, "paste-ups" of the legislation and a history of the legislation are maintained to ensure that commencements with unknown dates are noted. Making a history notation when the date becomes known is sufficient for the legislation to be recognised as having commenced.</p> <p>For example, when a Principal Act is commenced by proclamation, the Records Clerk notes the proclamation date on the "paste-up" and the Executive Officer notes the proclamation date in the bound annual volumes and continuing reprint.</p> <p>For Amendment Acts where the commencement is by proclamation the Records Clerk marks up the amendments on the "paste-ups" with a notation of TBP (To Be Proclaimed). Upon proclamation the Records Clerk strikes out the TBP notation and the Executive Officer updates the bound annual volumes and continuing reprint.</p>	<p>Legislation is loaded on to the consolidated database by a commencement utility. This utility uses the information entered into the commencement provision of the legislation enabling the legislation to be commenced either in full or in part. EnAct also utilises the commencement dates to provide consolidation of legislation that is in the legislation database. As the integrity of the legislation database is based on the commencement dates, the process needs to be completed accurately.</p> <p>The utility which is provided to commence legislation, operates independently to the workflow "drafting files".</p> <p>EnAct also provides the facility to produce commencement reports detailing legislation that has commenced and legislation yet to be commenced.</p>	<p>Who will process commencements and how?</p> <p>How will a drafter be able to determine which legislation is still to be commenced and when?</p>
Expiry of Legislat'n	<p>Legislation may be expired either in full or in part. Statutory Rules expire by default after 10 years, if not expired earlier. They can also be extended for an additional twelve (12) months under the Subordinate Legislation Act 1992.</p> <p>If the complete legislation is to be expired then the Records Clerk removes the entire "paste-up" of the legislation and the Executive Officer strikes out the legislation from the bound annual volume.</p> <p>If parts of the legislation are to be expired then the Records Clerk notes on the "paste-ups" and on the history the expiry date of each affected provision.</p>	<p>EnAct utilises the expiry dates to provide consolidation of legislation in the database. The expiry of legislation occurs in a step outside of the workflow for each "drafting file". The step needs to be initiated by a nominated person(s).</p> <p>Expiry of legislation is processed by the same procedure as the commencement of legislation (as described in Section Commencement of Legislat'n).</p> <p>EnAct provides the capability to record information relating to the expiry of a complete piece of legislation or parts of the legislation. It is necessary for each piece of legislation to be expired by completing the appropriate step. As the</p>	As above

	The Executive Officer maintains an electronic list of expiry dates.	integrity of the legislation database is based on the accuracy of expiries, the process needs to be completed accurately.	
Repeal / Rescind / Revoke Legisl'n	Legislation may be repealed, rescinded or revoked. For convenience, future reference to these processes within this document will be made by using the word "repeal" to mean repeal or rescind or revoke, as appropriate. If the complete legislation is to be repealed then the Records Clerk removes the entire "paste-up" of the legislation and the Executive Officer strikes out the legislation from the bound annual volume. If parts of the legislation are to be repealed then the Records Clerk notes on the "paste-ups" and history the date of repeal for each affected provision. The Executive Officer maintains an electronic list of repeal dates.	Repeal, rescind and revoke tasks can only be performed by creating a Change Description Document (CDD).	
Amend'ts to Legislat'n Not on Database	Amendments to legislation are processed as per amending bills.	<p>EnAct will provide access to all legislation, Principal Acts, Amendment Acts and Statutory Rules that are on the database. Amending principal legislation on the database involves obtaining a copy of the Principal Act to be amended (at a selected point in time), drafting the amendments required, creating a "stub" and merging the two together. However, if the Principal Act has not been converted and added to the legislation database then this process cannot be performed as per amending bills.</p> <p>This issue needs to be addressed by the Legislation Conversion and Maintenance (LCM) report before EnAct is implemented.</p> <p>The system will allow for amendments to be made to principal legislation that is not currently on the database using a by-pass mechanism. The by-pass mechanism allows for the preparation of amendments in the current manner.</p> <p>Full consolidation of legislation is not possible until all Statutory Rules, Acts and Amendment Acts are on the database.</p>	<p>Who will decide if legislation not currently on the database is added to it before new amendments are added to it?</p> <p>How will records be kept for amendments not on the database?</p> <p>Who will determine if the by-pass mechanism will be used and how?</p>
Commence ment & Expiry of Unconverted Legislat'n	<p>Unconverted legislation is the legislation that was not converted into SGML to load into the authorised and consolidated legislation database, as part of the Data Conversion Project (DCP). The DCP was established to consolidate and convert Tasmanian legislation into electronic SGML tagged documents suitable for use in EnAct.</p> <p>Two Statutory Rules and all Acts passed by Parliament up to 1 February 1997 were to be converted. This means that when EnAct is implemented there will be legislation that has not been converted. (ie no current processes)</p>	<p>EnAct will provide access to all legislation, Principal Acts, Amendment Acts and Statutory Rules that are on the database. The commencement and expiry procedure for legislation needs to be initiated after the legislation has been saved to the legislation system database. Therefore, if the legislation has not been converted and added to the legislation database then this process cannot be performed as per the Commencement of Legislat'n or the expiry of legislation.</p> <p>This issue needs to be addressed by the Legislation Conversion and Maintenance (LCM) report before EnAct is implemented.</p> <p>Full consolidation of legislation is not possible until all Statutory Rules, Acts and Amendment Acts are on the database.</p>	<p>When and how will this legislation be converted to the database after the system is formally launched?</p> <p>Who will decide if unconverted legislation, which has expired prior to being converted, will or will not be added to the database?</p>

Maint'g Paper Copies of Legislat'n	Annual volumes of Acts contain copies of all Acts that were passed by Parliament in a calendar year. Annual volumes of Statutory Rules contain copies of all Statutory Rules that were notified in the Gazette in a calendar year. They also contain tables of disallowed Statutory Rules, Attorney General exemptions and rulings and Statutory Rules arranged under Governing Acts. The annual volumes are prepared by the OPC as special Camera Ready versions, which includes an index and consecutive page numbers on every page. This version is provided to the Government Printer for printing, binding and distribution. The annual volumes are maintained for research and reference purposes.	The drafters within the OPC have access to Information Retrieval which allows them to search the authorised and consolidated database of all legislation. This includes Principal Legislation and amendments. A drafter can obtain a copy of consolidated legislation at a specified date ie. current date, a date in the future or a date in the past. Additionally, a copy of the authorised and consolidated database will be available to any of the OPC staff via the WWW.	Tasks basically automated, though the roles of the individuals who had responsibility for these tasks will probably only change, rather than become redundant, due to the shortage of knowledgeable people in this area and the high workload of the office.
Paste-ups	To maintain a current copy of amended legislation, paper copies of the Principal Legislation are manually updated by: "pasting" the additional or changed provisions to the legislation; making editing marks on the legislation for minor changes to wordings; striking out provisions which have been repealed or have expired; and removing complete paste-ups of Acts/Statutory Rules that have been repealed or have expired. The paste-ups are maintained for research and reference purposes.	Not required	As above.
Pamphlet copies of Legislat'n	All legislation that has been passed by Parliament or notified in the Gazette is printed in pamphlet form by the Government Printer. These paper copies of the legislation are printed on A4 size paper. These are maintained for research and reference purposes.	Status undetermined at time of writing	As above.
Pamphlet copies of Bills	Not all Bills which are introduced to Parliament become Acts. Copies of all Bills which have been introduced in Parliament are maintained in a paper form. These are maintained for research and reference purposes.	All legislation which has been drafted using the new legislation system will be available to the OPC. The drafting file remains on the OPC database, but does not become authorised legislation and therefore will not be retrievable in the legislation database. EnAct allows for the CPC to re-assign a "drafting file" to another drafter and to modify the status of a "drafting file" to indicate that a file is on hold or cancelled. Also, documents within drafting files can be shared by making copies of files "available" to other drafters.	As above.
Dept of Justice	The OPC forwards the following to the Department of Justice of copies of all "paste-ups" of legislation (Acts and Statutory Rules) and the signed copy of each Statutory Rule.	All authorised legislation will be maintained in the legislation database. Anyone with access to the WWW will be able to access this information and obtain consolidated copies of all authorised legislation, at a specified date.	As above.
Maint'g Indexes	The Records Clerk maintains a number of indexes. These indexes comprise, but are not restricted to: Bill numbers; Statutory Rule numbers; Act numbers; all Acts; all Principal Acts; administrative summaries for	EnAct has the ability to provide the information currently maintained in indexes. This is achieved by workflow reports which utilise the information captured in the legislation database through the following utilities:	As above.

	<p>all Acts; all Bills; amendments to legislation currently assigned; Statutory Rules made under each Act; subordinate legislation exempted from being or ruled to be a Statutory Rule; work assignment within the OPC for Bills; work assignment within the OPC for Statutory Rules; master list of Indexes to Legislation of Tasmania; commencement dates of whole/parts of legislation; expiry dates of whole/parts of legislation; cross references contained in Acts; Statute Law Revision and reprint changes; Amendment Acts to be repealed; and Footnotes.</p>	<ul style="list-style-type: none"> • a tool that allows for the legislation on the database to be searched using different criteria - refer Section Drafting Research; • assigning Bill numbers according to the next available number - refer Section; • assigning Statutory Rule numbers according to the next available number; • Act numbers are applied to the legislation when it is commenced; • a commencement function which allows for parts of the legislation to be commenced on different dates; and • an expiry function which allows for parts of the legislation to be expired on different dates. 	
--	--	--	--

New processes resulting from the LSP cover issues such as ongoing training and documentation, system security, system administration, IT support, procedures for handling EnAct problems and issues and changes to the EnAct system.

Appendix 4: Insert/Omit and Strikethrough/underline styles

Principal Act

When polling may be adjourned

137 -(1)...

...

(5) Where polling for an election at one or more polling-booths in any Assembly subdivision or Assembly subdivisions, or polling for an election at one or more polling-booths in a Council division, has been adjourned, only those electors who are enrolled for that Assembly subdivision or those Assembly subdivisions or, as the case may be, that Council division, and who have not already voted at that election, are entitled to vote at the adjourned poll.

Amendment Act- insert/omit style

Section 137 amended (When polling may be adjourned)

31 Section 137 (5) of the Principal Act is amended by omitting "subdivision" and "subdivisions" (each twice occurring) and substituting "division" and "divisions" respectively.

Amendment Act- strikethrough/underline style

Section 137 amended (When polling may be adjourned)

31 Section 137 (5) of the Principal Act is amended by omitting all text that has been struckthrough and inserting all text that has been underlined, as follows:

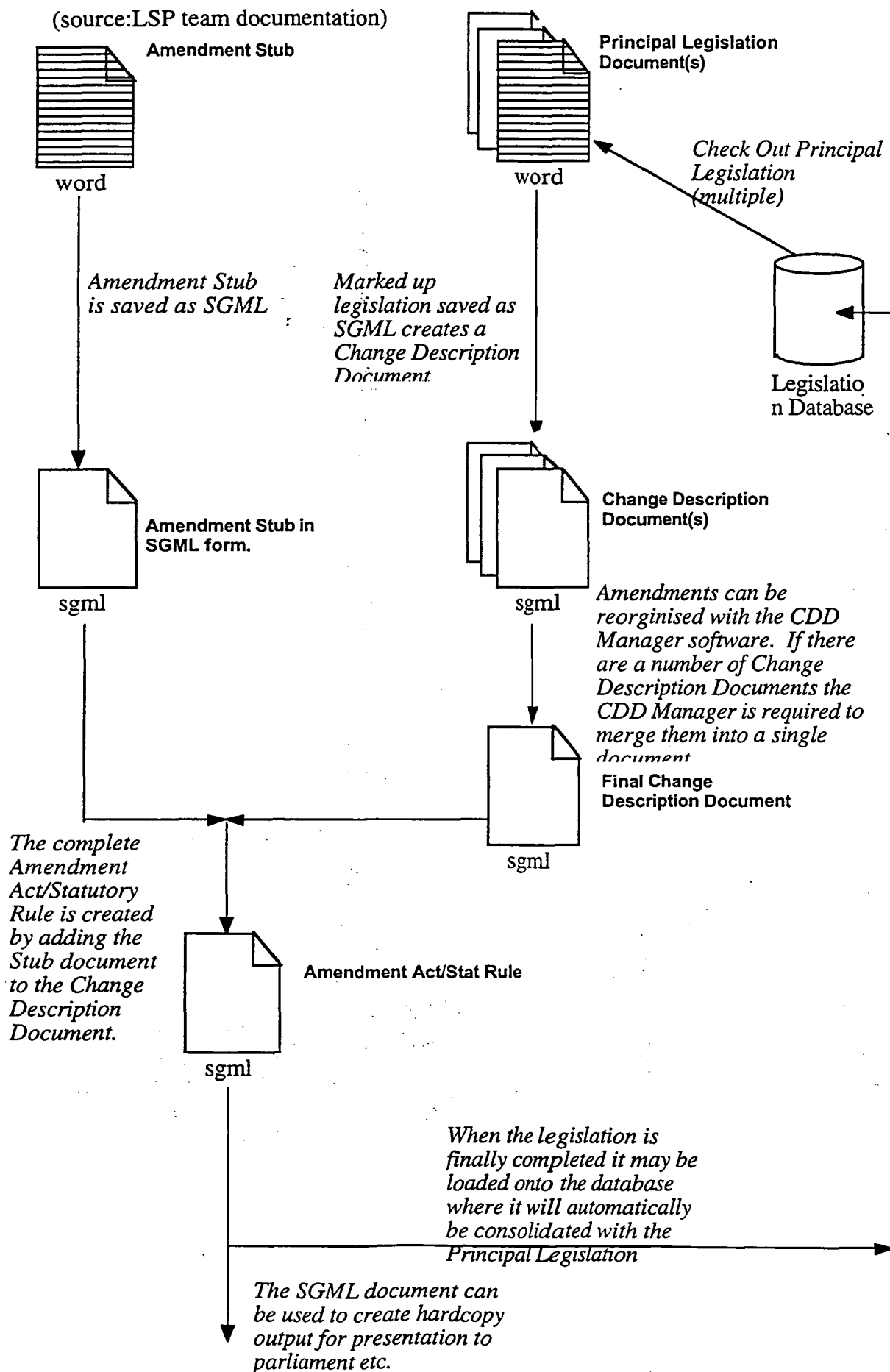
Where polling for an election at one or more polling-booths in any Assembly division ~~subdivision~~ or Assembly division ~~subdivisions~~, or polling for an election at one or more polling-booths in a Council division, has been adjourned, only those electors who are enrolled for that Assembly division ~~subdivision~~ or those Assembly division ~~subdivisions~~ or, as the case may be, that Council division, and who have not already voted at that election, are entitled to vote at the adjourned poll.

Source: LSP project documentation

Appendix 3

Broad conceptual overview of the EnAct system

(source:LSP team documentation)



Bibliography

- Abrahamson, E. (1991). "Managerial fads and fashions: the diffusion and rejection of innovations." *Academy of Management Review* 16(3): 586-612.
- Ackroyd, S. (1992). Paradigms Lost: Paradise Regained? *Rethinking Organization: New Directions in Organizational Theory and Analysis* Eds. M. Reed and M. Hughes. London, Sage. 102-119.
- Allen, T. J. and O. Hauptman (1994). The Influence of Communication Technologies on Organisational Structure: A Conceptual Model for Future Research. *Information Technology and the Corporation of the 1990s: Research Studies* Ed. T. J. Allen and M. S. Scott Morton. Oxford, Oxford University Press. 475-483.
- Aldrich, H. E. (1992). Incommensurable Paradigms? Vital Signs from Three Perspectives. *Rethinking Organization: New Directions in Organizational Theory and Analysis* Eds. M. Reed and M. Hughes. London, Sage. 17-45.
- Alter, S. (1996). *Information Technology- a management perspective*. USA, Addison Wesley.
- Amoako-Gyampah, K. and K. B. White (1993). "User involvement and user satisfaction: a exploratory contingency model." *Information and Management* 25: 1-10.
- Andrews, R. (1993). *Columbia Dictionary of Quotations*. Columbia University Press.
- Applegate, L. M. and King, J. L. (26/5/97). *Crisis in the Case Study Research*.
<http://www.hbs.edu/applegate/research/case.html>.
- Ashmos, D. P. and G. P. Huber (1987). "The systems paradigm in organization theory: correcting the record and suggesting the future." *Academy of Management Review* 12(4): 607-621.
- Atkinson, P. and Hammersley, M. (1994) "Ethnography and Participant Observation" NK Denzin and YS Lincoln, *Handbook of Qualitative Research* London, Sage. 248-261.
- Aungles, S. and S. Parker (1992). *Work, Organisations and Change: Themes and Perspectives in Australia*. St Leonards, NSW, Allen and Unwin.
- Australia, Parliament. (Melham, Daryl (chairman) (1993). House of Representatives Standing Committee on Legal and Constitutional Affairs: Clearer Commonwealth Law. Parliament of the Commonwealth of Australia.
- Avison and Fitzgerald (1995). *Information Systems Development - Methodologies, Techniques and Tools*. Blackwell.
- Banbury, J. (1987). Towards a framework for systems analysis practice. *Critical issues in information systems research* Ed. J. Bolland and R. Hirschheim. John Wiley and Sons. 79-96.
- Bansler, J. and K Bødker (1993). "A reappraisal of structured analysis in an organisational context." *ACM Transactions on Information Systems* 3 (2) pp 163-193
- Barley, S. R. (1990). "The Alignment of Technology and Structure through roles and networks." *Administrative Science Quarterly* 35(March): 61-103.
- Barley, S. R. (1995). Images of Imaging: Notes on Doing Longitudinal Fieldwork. *Longitudinal Field Research Methods: Studying Processes of Organisational Change* Eds. G. P. Huber and A. H. Van de Ven. London, Sage. 1-37.
- Barley, S. R. and P. S. Tolbert (1997). "Institutionalization and Structuration: Studying the Links between Action and Institution." *Organization Studies* 18(1): 93-117.
- Barrett, M. and G. Walsham (1995). "Managing IT for Business Innovation: Issues of Culture, Learning, and Leadership in a Jamaican Insurance Company." *Journal of Global Information Management* 3(3): 25-33.
- Barrett, M. and G. Walsham (1995). "Using IT to Support Business Innovation: A Case Study of the London Insurance Market." *Scandinavian Journal of Information Systems* 17(2): 1995.

- Bashein, B.J and M.L. Markus "A credibility equation for IT specialists" *Sloan Management Review* Summer 1997. pp 35-44.
- Bell, D., I Murrey, J Pugh, (1992)*Software Engineering: a programming approach* 2nd ed, Hamel Hempstead, Prentice Hall
- Becker, F. D. (13/1/1997). Change Management. <http://wsp.human.cornell.edu/Change>.
- Benjamin, R. (1993). Managing Information Technology Enabled Change. *Human, Organisational and Social Dimensions of Information Systems Development* Ed. D. Avison, J. E. Kendell and J. I. DeGross. North-Holland, Elsevier Science Publishers (IFIP). 381-397.
- Benjamin, R. I. and E. Levinson (1993). "A Framework for Managing IT-Enabled Change." *Sloan Management Review* 1993(Summer): 23-33.
- Berger, L. (1994). Change Management. *The Change Management Handbook: A road map to corporate transformation* Ed. L. A. Berger, M. J. Sikora and D. R. Berger. Burr Ridge, Illinois, Irwin Professional Publishing. 3-23.
- Berger, P. and T. Luckman (1966). *The Social Construction of Reality*. Hammondsworth, Penguin.
- Berlin, I. (1969). Two Concepts of Liberty. *Four Essays on Liberty* . I. Berlin. London, Oxford University Press. 118-172.
- Bernstein, R. J. (1989). Social theory as critique. *Social Theory of Modern Societies: Anthony Giddens and his Critics* Eds. D. Held and J. B. Thompson. Cambridge, Cambridge University Press. 19-33.
- Berry, D. (1997) "Why Legislative Drafting Services should not be privatised". *The Loophole* March 1997. pp 52-53
- Bijker, W. E. (1989). The social construction of Bakelite: Toward a theory of Invention. *The Social Construct of Technological Systems* Eds. W. E. Bijker, T. P. Hughes and T. Pinch. Cambridge Massachusetts, The MIT Press.
- Blackler, F. (1992). Formative Contexts and Activity Systems: Postmodern approaches to the management of change. *Rethinking Organization: New Directions in Organizational Theory and Analysis* Ed. M. Reed and M. Hughes. London, Sage. 273-294.
- Boehm, B. W. (1988). "A Spiral Model of Software Development and Enhancement." *Computer* 1988(May): 61-72.
- Boland, R. J. J. and R. H. Greenberg (1992). "Method and Metaphor in Organisational Analysis." *Accounting Management and Information Technology* 2(2): 117-141.
- Borum, F. and J. Christianssen (1993). "Actors and Structure in IS projects: what makes implementation happen." *Scandinavian Journal of Management* 9(1): 5-28.
- Brinckman, H. (1991). Technological and Organisational Innovation. *Information System, Work and Organization Design* Ed. P. Van den Besselaar, A. Clement and P. Jarvinen. North-Holland, Elsevier Science Publishers (IFIP).
- Brown, A. (1995). "Managing understandings: politics, symbolism, niche marketing and the quest for legitimacy in IT implementation." *Organization Studies* 16(6): 951-969.
- Brussard, B. K. (1988). "Information Resource Management in the Public Sector." *Information and Management* 15: 85-92.
- Bryant, C. and D. Jary (1991). Introduction: Coming to terms with Anthony Giddens. *Giddens's Theory of Structuration: a critical appreciation* Eds. C. Bryant and D. Jary. London, New York, Routledge. 1-31.
- Brynjolfsson, E. (1993). "The Productivity Paradox of Information Technology." *Communications of the ACM* 36(12):
- Buchanan, D. (1993). "Review of *A Strategy of Change: Concepts and Controversies in the Management of Change* by David Wislon, 1992." *Journal of Management Studies* 30(3): 684-686.
- Buchanan, D. and D. Boddy (1992). *The expertise of the change agent: public performance and backstage activity*. Hempstead, UK, Prentice Hall.

- Burch, J. G. (1992). *Systems Analysis, Design and Implementation*. Boston, Boyd and Fraser.
- Burch, J. and G. Grudnisky (1986). *Information Systems: Theory and Practice*. New York, John Wiley and Sons.
- Burn, J. M. (1989). "The impact of Information Technology on Organisational Structures." *Information and Management* 16: 1-10.
- Burnes, B. (1996). *Managing Change: A Strategic Approach to Organisational Dynamics*. London, Pitman Publishing.
- Burrell, G. (1992). Back to the future: time and organization. *Rethinking Organization: New Directions in Organizational Theory and Analysis* Ed. M. Reed and M. Hughes. London, Sage. 165-183.
- Burrell, G. and G. Morgan (1979). *Sociological Paradigms and Organisational Analysis*. London, Heinemann.
- Burris, B. (1993). *Technocracy at work*. NY, State University of New York Press.
- Campbell, B. T. (1984). Information Technology and Civil Service Culture. *The Computer Revolution in Public Administration: The Impact of IT in Government* Ed. D. C. Pitt and B. C. Smith. London, Wheatsheaf. 73-89.
- Carnall, C. A. (1990). *Managing Change in Organisations*. NY, Prentice Hall.
- Cavaye, A. and P. Cragg (1994). *A success factors model for inter-organisational systems*. Business process reengineering: information system opportunities and challenges, Gold Coast, Qld Australia, North Holland.
- Centeno, M. A. (1993). "The new Leviathan: The dynamics and limits of technocracy." *Theory and Society* 22: 307-335.
- Chartrand, R. L. and R. C. Ketcham (1994). "Opportunities for the Use of Information Resources and Advanced Technologies in Congress: A Study for the Joint Committee in the Organization of Congress (A Consultant Report)." *The Information Society* 10: 181-222.
- Checkland, P. B. (1988). "Soft Systems Methodology: An Overview." *Journal of Applied Systems Analysis* 15: 27-30.
- Checkland, P. and S. Howell (1993). "Information Management and organisational processes: an approach through soft systems methodology." *Journal of Information Systems* 3(1): 5-16.
- Checkland, P. and J. Scholes (1990). *Soft Systems Methodology in Action*. Chichester, UK, John Wiley and Sons.
- CHIPS, (. H. R. I. a. P. S. P. (1996). How we did it. Government of British Columbia.
- Clegg, C. et al. (4/11/96). Systems Concepts Ltd: The performance of Information Technology and the role of human and organisational factors. http://www.system_concepts.com/stds/clegg.html#RTFToC2.
- Collins, D. (1996). "New paradigms for change? Theories of organization and the organisation of theories." *Journal of organisational change management* 9(4): 9-23.
- Cook, J. (1995). "Legislation conserved in natural State." *Perspectives* (34):
- Cooper, R. B. (1994). "The inertial impact of culture on IT implementation." *Information and Management* 27: 17-31.
- Coopers&Lybrand (13/1/1997). Government Consulting - Change Management. <http://www.colybrand.com/clc.gov/cm.html>.
- Craig, J. and P. Yetton (1992). "Business Process Redesign: A Critique of *Process Innovation* by Thomass Davenport." *Australian Journal of Management* 17(2): 285-306.
- Craig, J. F. and P. W. Yetton (1994). *Top down and bottom up management of BPR*. Business process reengineering: information system opportunities and challenges, Gold Coast, Qld Australia, North Holland.
- Crowe, M., R. Deeby, et al. (1996). *Constructing Systems and Information: A process view*. McGraw Hill.

- Curtis, B., H. Krasner, et al. (1988). "A Field Study of the Software Design Process for Large Systems." *Communications of the ACM* 31(11): 1268-1287.
- Dachouffe, N. and R. LeSuisse (1989). OIDA: Office Information Systems Design Aids. *Office Information Systems: The Design Process* Ed. B. Pernici and V.-S. A.A. North Holland, Elsevier Science Publishers. 23-?
- Daft, R. C. (1995). *Organization theory and design*. Minneapolis/ St Paul, West Publishing Company.
- Darke, P. (1995). *Viewpoint development for requirements definition: towards a conceptual framework*. 6th Australasian conference on information systems, Perth, School of information systems, Curtin University.
- Darke, P. and G Shanks (1997). "Managing User Viewpoints in Requirements Definition" *8th Australian Conference of Information Systems*. Adelaide, September 1997. 495-508.
- Davenport, T. E. and J. E. Short (1990). "The New Industrial Engineering: Information Technology and Business Process Redesign." *Sloan Management Review* 1990(Summer): 11-27.
- Davenport, T. H. and D. B. Stoddard (1994). "Reengineering: Business Change of Mythic Proportions?" *MIS Quarterly* 1994(June): 121-127.
- Davidow, R. (1994). Structuring the Change Initiative. *The Change Management Handbook: A road map to corporate transformation* Ed. L. A. Berger, M. J. Sikora and D. R. Berger. Burr Ridge, Illinois, Irwin Professional Publishing. 133-137.
- Davies, L. and P. Ledington (1991). *Information in Action: Soft Systems Methodology*. London, MacMillan.
- Davis, G. B. (1992). Systems Analysis and Design: A research strategy macro-analysis. *Challenges and Strategies for Research in Information Systems Development* Ed. W. Cotterman and J. A. Senn. Chichester, UK, John Wiley and Sons. 365-414.
- Dawson, P. (1994). *Organisational Change - A Processual Approach*. London, P Chapman.
- Dawson, P. (1997). "Beyond conventional change models: a processual perspective." *Asia Pacific Journal of Information Systems* 34(2): 57-70.
- DDS (13/1/1997). DDS Change Management. <http://www.oldsinc.com/Change.htm>.
- DeCock, C. (1995). "A rejoinder to and reply from Weaver and Gioia." *Organization Studies* 16(4): 699-704.
- Dewitz, S. D. (1996). *Systems Analysis and design and the transition to objects*. New York, McGraw Hill.
- Dey, I. (1993). *Qualitative Data Analysis: A User friendly guide for social scientists*. London, Routledge.
- Dickerson, R. (1954) *Legislative Drafting*, Boston, Little Brown and Company
- Dillard, J. F. and K. Yuthas (1997). *Fluid structures: a structuration approach to evaluating information technology*. *Advances in Accounting Information Systems*,
- Eason, K. (1988). *Information Technology and Organisational Change*. London, New York, Philadelphia, Taylor and Francis.
- Eliason, A. L. (1990). *Systems Development: Analysis, Design and Implementation*. USA, Scott, Foresman and Company.
- Elmore, R. F. (1979). "Backward Mapping: Implementation Research and Policy Decisions." *Political Science Quarterly* 94: 601-616.
- Elkjaer, B., P. Flensburg, et al. (1991). "The commodification of expertise: the case of systems development consulting." *AMIT* 1(2): 139-156.
- Elster, J. (1983). *Explaining Technological Change: A Case Study in the Philosophy of Science*. Cambridge and Oslo, Cambridge University Press and Universitetsforlaget.
- Evans, R. (1996) "The hand that signed the paper: Are legislative drafters behind the times?" *Law Institute of Victoria* April 1996 pp 12-14.

- Fayol, H. (1990). General principles of management. *Organization theory* Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 181-202.
- Fetterman, D. M. (1991). A Walk through the wilderness: learning to find your way. *Experiencing Fieldwork: An Inside View of Qualitative Research* Eds. W. B. Shaffir and R. A. Stebbins. Newbury Park, Sage.
- Fitzgerald, B. (1996). "Formalised systems development methodologies: a critical perspective." *Information systems journal* 6: 3-23.
- Floyd, C., F. Reisin and G. Schmidt (1989). "STEPS to software development with users" 2nd European Software Engineering Conference, University of Warwick, UK, September 1989. 45-64.
- Franz, C. R. and D. Robey (1987). Strategies for Research on Information Systems in Organizations: A Critical Analysis of Research Purpose and Time Frame. *Critical Issues in Information Systems Research* Eds. R. J. J. Boland and R. A. Hirschheim. Chichester UK, John Wiley and Sons. 205-225.
- Frenzel, C. W. (1996). *Management of Information Technology*. Boston, Boyd and Fraser.
- Friedman, A. L. and D. S. Cornford (1989). *Computer Systems Development: History, Organisation and Implementation*. NY, John Wiley and Sons.
- Gasson, S. and N. Holland (1995). *The Nature and Process of IT-Related Change*. Information Technology and Changes in Organisational Work, Cambridge, Chapman and Hall.
- Giddens, A. (1979). *Central Problems in Social Theory: Action, structure and contradiction in social analysis*. London, The Macmillan Press Ltd.
- Giddens, A. (1989). A reply to my critics. *Social Theory of Modern Societies: Anthony Giddens and his Critics* Ed. D. Held and J. B. Thompson. Cambridge, Cambridge University Press. 235-248.
- Giddens, A. (1991). Structuration theory: past, present and future. *Giddens's Theory of Structuration: a critical appreciation* Eds. C. Bryant and D. Jary. London, New York, Routledge. 201-221.
- Gioia, D. A. and E. Pitre (1990). "Multiparadigm perspectives on Theory Building." *Academy of Management Review* 15(4): 584-602.
- Goldkuhl, G. and A. Rostlinger (1993). Joint Elicitation of Problems: Important Aspect of Change Analysis. *Human, Organisational and Social Dimensions of Information Systems Development* Ed. D. Avison, J. E. Kendall and J. I. DeGross. North-Holland, Elsevier Science Publishers (IFIP). 107-125.
- Gray, B. and S. S. Ariss (1985). "Politics and Strategic Change Across Organizational Life Cycles." *Academy of Management Review* 10(4): 707-723.
- Greenleaf, G. (1994). "Computerised Australian legislation - the state of play." *The Australian Law Journal* 68(March): 231-233.
- Greenleaf, G.W., A.S. Mobray, D.P. Lewis (1988) *Australian Computerised Legal Information Handbook* Sydney, Butterworths
- Greenleaf, G., A. Mobray, et al. (1995). "Public Access to Law Via Internet: The Australasian Legal Information Institute." *JLIS* 6(1): 49-69.
- Gregson, N. (1989). On the (ir) relevance of structuration theory to empirical research. *Social Theory of Modern Societies: Anthony Giddens and his Critics* Eds. D. Held and J. B. Thompson. Cambridge, Cambridge University Press. 235-248.
- Griener, L. E. (1967). "Patterns of Organisation Change." *Harvard Business Review* (May-June): 119-130.
- Hackett-Jones, G. (1988). The scar of Odysseus and the role of parliamentary counsel in the legislative process. *Essays on Legislative Drafting* Ed. D. S. L. Kelly. Adelaide, Adelaide Law Review Association. 43-56.
- Hamel, J., S. Dofour, et al. (1993). *Case Study Methods*. Newbury Park, Calif., Sage.
- Hammer, M. (1990). "Reengineering Work: Don't Automate, Obliterate." *Harvard Business Review* July-August: 104-112.

- Hammer, M. and J. Champy (1990). *Reengineering the Corporation*. New York, NY, Harper Business.
- Hammer, M. and S. A. Stanton (1995). *The Reengineering Revolution: A Handbook*. Sydney, Harper Business.
- Han, C. K. (1994). "Structuration theory and Strategy Research: A Case study of National Information Technology Policy Formation in Malaysia." *Malaysian Journal of Management Science* 3(2): 75-97.
- Harrington, J. (1991). *Organisational Structure and Information Technology*. Hempstead, UK, Prentice Hall International.
- Hartzel, K. S. and P. R. Flor (1995). *Expectation Formation in the Information System Development Process*. Proceeding, American Conference of Information Systems.
- Harvey, L.J. and M.D. Myers. (1995) "Scholarship and practice: the contribution of ethnographic research methods to bridging the gap", *Information Technology and People* 8(3). pp 13-27.
- Hawkes, G. (1994). "Could you repeat that in English?" *Gazette*
- Hawryszkiewicz, I. (1994). *Introduction to systems analysis and design*. Sydney, Prentice Hall.
- Heifetz and Halle (13/1/1997). HHCG Change Management. <http://www.leading-change.com/>.
- Held, D. and J. B. Thompson (1989). Editors' Introduction. *Social Theory of Modern Societies: Anthony Giddens and his Critics* Eds. D. Held and J. B. Thompson. Cambridge, Cambridge University Press. 1-18.
- Highsmith, J. (1997) "Messy Exciting and Anxiety-Ridden Adaptive Software Development." *American Programmer* 1997 (April). 23-29
- Hirschheim, R. (1985). Information Systems Epistemology: An Historical Perspective. *Research Methods in Information Systems* Ed. E. Mumford. North-Holland, Elsevier.
- Hirschheim, R. and H. K. Klein (1989). "Four Paradigms of Information Systems Development." *Communications of the ACM* 32(10): 1199-1216.
- Hirschheim, R., H. K. Klein, et al. (1991). "Information Systems Development as Social Action: Theoretical Perspective and Practice." *OMEGA International Journal of Management Science* 19(4): 587-608.
- Hirschheim, R. and H. Klein K (1992). "Paradigmatic influences on information systems development methodologies: evolution and conceptual advances." *Advances in computers* 33: 293-391.
- Hirschheim, R. and H. K. Klein (1992). A Research Agenda for Future Information Systems Development Methodologies. *Challenges and Strategies for Research in Information Systems Development* Eds. W. Cotterman and J. A. Senn. Chichester, UK, John Wiley and Sons. 235-256.
- Hirschheim, R., H. Klein, et al. (1994). *Control, Sensemaking and Argumentation: Articulation and Exploring the Intellectual Structures of Information Systems*. Australian Conference of Information Systems, Melbourne, Monash University Printing Services.
- Hirschheim, R. and H. K. Klein (1994). "Realizing Emancipatory Principles in Information Systems Development: The Case for ETHICS." *MIS Quarterly* 1994(March): 83-109.
- Hirschheim, R. and M. Newman (1991). "Symbolism and Information Systems Development: Myth, Metaphor and Magic." *Information Systems research* 2(1): 29-62.
- Hocking, L. J. and C. D. Keen (1995). *An assessment of a centralised IT policy making body based on forward and backward mapping*. PACIS, Singapore,
- Hoffer, J. A., J. F. George, et al. (1996). *Modern Systems Analysis and Design*. Menlo Park, CA, The Benjamin/Cummings Publishing Company Inc.
- Hull, C. (1994) "Improving public access to legislation database". *Canberra Times*. 30 May 1994 p 17
- Iivari, J. (1992). "The organisational fit of information systems." *Journal of Information Systems* 2(1): 3-28.
- Iivari, J. and R. Hirschheim (1996). "Analyzing Information Systems Development: A comparison and analysis of eight IS development approaches." *Information Systems* 21(7): 551-575.

Information Strategy Unit, Department of Premier and Cabinet, Tasmania (1997) *Guidelines for Project Management* (Unpublished).

Integration, P. S. (13/1/1997). Cultural change management.
[http://www.glam.ac.uk/pages/staff/ajcblyth/Teaching/Change-management/change management.html](http://www.glam.ac.uk/pages/staff/ajcblyth/Teaching/Change-management/change%20management.html).

International Organization for Standardization (ISO) (1986). *Information processing – Text and Office Systems – Stand Generalized Markup Language*. Geneva, ISO

Jackson, M. (1992). "An integrated programme for critical thinking in information systems research." *Journal of Information Systems* 3(2): 83-95.

Jackson, N. and P. Carter (1991). "In defence of paradigm incommensurability." *Organization Studies* 21(1): 109-127.

Jenner, M. (1993). *Managing for Quality*. APT Quality Management.

Jones, M. (1995) "Organisational learning: collective mind or cognitivist metaphor?" *Accounting, Management and Information Technology* 5 (1) pp 61-77.

Jones, M. and J. Nundakumar (1993). *Structured Development? A structurationist analysis of the development of an executive information system*. Cambridge University.

Jones, M. and G. Walsham (1992). The Limits of the Knowable: Organisational and Design Knowledge in Systems Development. *The Impact of Computer Supported Technologies on Information Systems Development* Ed. K. E. K. e. al. North Holland, Elsevier Science Publishers. 195-213.

Jordan, E. W. and J. J. Machesky (1990). *Systems Development: Requirements, Evaluation, Design, and Implementation*. Boston, PWS-Kent.

Jorgensen, D. L. (1989). *Participant Observation: A methodology for Human Studies*. Newbury Park, Sage Publications.

Joshi, K. (1991) "A model of users' perspectives on change: the case of information systems technology implementation." *Management Information Systems Quarterly* 1991 (June) 228-242.

Kaasboll, J. (1995). *Knowledge from the inside and outside in participative development and research on participative development*. 6th Australasian conference on information systems, Perth, School of information systems, Curtin University.

Kautz, K. (1995). *Information Technology Transfer and implementation: The introduction of an electronic mail system in a public service organization*. Diffusion and Adoption of Information Technology, Oslo, IFIP WG 8.6.

Kautz, K. and T. McMaster (1994). "Introducing Structured Methods: An Undelivered Promise? - A case study" *Scandinavian Journal of Information Systems* 6 (2) 59-78.

Keen, P. (1981). "Information Systems and Organisational Change." *Communications of the ACM* 24(1): 24-33.

Keen, P. (1991). *Shaping the Future: Business Design through Information Technology*. USA, Harvard Business School Press.

Keen, P. P. W. (1991). Relevance and Rigor in Information Systems Research: Improving Quality, Confidence, Cohesion and Impact. *Information Systems Research: Contemporary Approaches and Emergent Traditions* Eds. H.-E. Nissen, K. H.K. and R. Hirschheim. North Holland, Elsevier Science Publishers (IFIP). 27-49.

Kendall, K. and J. Kendall (1992). *Systems Analysis and Design*. Englewood cliffs, Calif, Prentice Hall International Editions.

Khushalani, A. R. Smith and S. Howard (1994). "What happens when designers don't play by the rules: towards a model of opportunistic behaviour in design." *Australian Journal of Information Systems* 1 (2) 13-31.

Kimberly, J. R. (1976). "Issues in the Design of Longitudinal Organizational Research." *Sociological Methods and Research* 4(3): 321-347.

- Klein, H. K. and R. Hirschheim (1993). The Application of Neohumanist Principles in Information Systems Development. *Human, Organisational and Social Dimensions of Information Systems Development* Eds. D. Avison, J. e. Kendell and J. I. DeGross. North Holland, Elsevier Science Publishers (IFIP). 263-280.
- Kling, R. (1987). Computerisation as an ongoing social and political process. *Computers and Democracy* Ed. G. Bjerknes, P. Ehn and M. Kyng. Aldershot, Avebury. 119-137.
- Kling, R. (1992). Behind the terminal: The Critical Role of Computing Infrastructure in Effective Information Systems Development and Use. *Challenges and Strategies for Research in Information Systems Development* Ed. W. Cotterman and J. A. Senn. Chichester, UK, John Wiley and Sons. 365-414.
- Kling, R. (1992). "Organizational Analysis in Computer Science." *The Information Society* 9: 71-87.
- Kling, R. (1994). Organisational Analysis and Organisational Informatics in Computer Science. *Encyclopedia of Computer Science*.
- Kling, R. and W. Scacchi (1982). "The Web of Computing: Computer Technology as Social Organisation." *Advances in Computers* 21: 2-60.
- Knudsen, T. (1993). Conception of Knowledge, Work Organization, Roles and Tasks in Three Systems Development Teams. Universitet i Bergen. Norway (working paper)
- Knudsen, T. (1993). The systems development process, organization structure and role of the development team. universitet i Bergen.
- Korac-Boisvert, N. and A. Kouzmin (1995). "IT development: methodology overload or crisis." *Science Communication* 17(1): 57-89.
- Korac-Boisvert, N. and A. Kouzmin (1995). "Transcending soft-core IT disasters in Public Sector Organisations." *Information Infrastructure and Policy* 4(1): 131-161.
- Kraemer, K. L. (1991). Strategic Computing and Administrative Reform. *Computerisation and Controversy: Value Conflicts and Social Choices* Ed. C. Dunlop and R. Kling. Boston USA, Academic Press Inc. 167-180.
- Krogstie, J. (1995). Conceptual Modelling for Computerised Information Systems Support in Organisations. Universitet i Trondheim, Norway (PhD dissertation)
- Kwon, T. H. and R. W. Zmud (1987). Unifying the fragmented models of information systems implementation. *Critical Issues in Information Systems Research* Ed. R. J. J. Boland and R. A. Hirschheim. Chichester UK, John Wiley and Sons. 227-251.
- Lacity, M. and R. Hirschheim (1993). *Information Systems Outsourcing*. Chichester, John Wiley and Sons.
- Land, F., P. Le Quesne, et al. (1992). Technology transfer: Organisational factors affecting implementation - some preliminary findings. *Challenges and Strategies in Systems Development* Ed. W. Cotterman and J. Senn. John Wiley and Sons. 65-81.
- Latour, B. (1995). *Social Theory and the Study of Computerised Work Sites*. IFIP..., Cambridge,
- Laudon, K. C. and J. P. Laudon (1995). *Information Systems: a problem solving approach*. Fort Worth, The Dryden Press.
- Lawrence, P. and J. Lorsch (1967). High performing organizations in three environments. *Organization theory* Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 76-94.
- Layder, D. (1993). *New Strategies in Social Research: An Introduction and Guide*. Cambridge UK, Polity Press.
- Lederer, A. L. and V. Sethi (1992). "Meeting the Challenges of Information Systems Planning." *Long Range Planning* 25(2): 69-80.
- Ledington, P. (1989). "The Interplay of Theory and Practice in the Ongoing Development of Soft Systems Thinking: An Illustration and Case Study from Sports Management." *Systems Practice* 2(1): 29-45.
- Legge, K. (1984). *Evaluating Planned Organisational Change*. Academic Press.
- Lewis, P. (1994). *Information Systems Development*. London, Pitman.

- Lim, A. (1993). Information Technology Support for Legislative Development: Tools for the legislative engineer. ANU, Faculty of Law.
- Lucas, H. C. J. (1981). *Implementation: The key to Successful Information Systems*. NY, Columbia University Press.
- Lyytinen, K. (1988). "Stakeholders, Information Systems Failures and Soft Systems Methodology: An Assessment." *Journal of Applied Systems Analysis* 15: 61-81.
- Madnick, S. E. (1991). The Information Technology Platform. *The Corporation of the Nineties* Ed. T. J. Allen and M. S. Scott Morton. Oxford, Oxford University Press.
- Majone, G. and A. Wildavsky (1978). "Implementation as Evolution." *Policy Studies Review Annual* 2: 103-117.
- Markus, M. L. (1983). "Power, Politics and MIS Implementation." *Communications of the ACM* 26(6): 430-444.
- Markus, M. L. and N. Bjorn-Anderson (1987). "Power over users: Its exercise by System Professionals." *Communications of the ACM* 30(6): 498-504.
- Martin, M. P. (1995). *Analysis and Design of Business Information Systems*. Englewood Cliffs, Prentice Hall.
- Martin, P. Y. and B. A. Turner (1986). "Grounded Theory and Organisational Research." *The Journal of Applied Behavioral Science* 22(2): 141-157.
- Mason, S. (1988). Law making, drafting and law reform. *Essays on Legislative Drafting* Ed. D. S. L. Kelly. Adelaide, Adelaide Law Review Association. 100-110.
- Mason D. and L. Wilcocks (1994) *Systems Analysis and Design* Henley on Themes, UK, Alfred Walker Ltd
- Mayo, E. (1949). Hawthorne and the Western Electric Company. *Organization theory* (1990) Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 345-357.
- McDonald, J. (1993) "How Legislation is initiated and developed" ISIS (Newsletter for the Office of Status for Women). Issue 10
- McGregor, D. (1960). Theory X and Theory Y. *Organization theory* (1990) Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 358-374.
- McKersie, R. B. and R. E. Walton (1991). Organizational Change. *The Corporation of the Nineties* Ed. T. J. Allen and M. S. Scott Morton. Oxford, Oxford University Press. 244-277.
- McLeod, G. and D. Smith (1996). *Managing Information TEchnology Projects*. Danvers, Boyd and Fraser.
- Merton, R. J. (1968). *Social Theory and Social Structure*. New York, The Free Press.
- Miles, M. B. and M. A. Huberman (1994). *Qualitative Data Analysis*. Thousand Oaks, Sage.
- Mintzberg, H. (1993). "The Pitfalls of Strategic Planning." *Calif Mgt Review* 36(1): 32-46.
- Mintzberg, H. (1994). *The Rise and Fall of Strategic Planning*. Hemel Hempstead, Prentice Hall.
- Mongomery, S. (1994). *Object-oriented information engineering: analysis, design and implementation*. Boston, AP Professional.
- Moody, D., G. Simsion, et al. (1995). *Stakeholder perspectives in conceptual modelling*. 6th Australasian conference on information systems, Perth, School of information systems, Curtin University.
- Moynihan, E. (1993). *Business Management and Systems Analysis* Henley-on-Themes, Alfred Walker Ltd.
- Mumford, E. (1995). *Effective Systems Design and Requirements Analysis*. London, UK, MacMillan.
- Murray, F. (1989). "The Organizational Politics of Information Technology: Studies from the UK Finacial Services Industry." *Technology Analysis and Strategic Management* 1(3): 1989.
- Myers, M. D. (1995). "Dialectical hermeneutics: a theoretical framework for the implementation of information systems." *Information Systems Journal* 5 1 pp 51-70:

- Myers, MD (1997). "Qualitative Research in Information Systems", IS WorldNet <http://www.auckland.ac.nz/msis/isworld/> accessed 10/12/97
- Myers, M.D. and L.W. Young (1997) "Hidden Agendas, power and managerial assumptions in information systems development", *Information Technology and People* 10(3) pp 224-240.
- Nandakumar, J. (1993). *The Practice of Executive Information Systems Development: An In-depth Study*. Cambridge.
- Nandakumar, J. and Jones, M. (1997) *Too close for comfort? Distance and engagement in interpretive information systems research*. *Information Systems Journal* 7 109-131
- Neuman, W. L. (1994). *Social Research Methods: Qualitative and Quantitative Approaches*. Boston, Allyn and Bacon.
- Noble, F. (1991). "Seven Ways to Develop Office Systems: A Managerial Comparison of Office System Development." *The Computer Journal* 34(2): 113-121.
- Nolan, R. L. (1979). "Managing the Crises in Data Processing." *Havard Business Review* 1979(Mar-Apr): 115-126.
- Nuttall, K. (1997) *Change Management and Facilitation Course* Researched and developed by Waterfield Consulting as part of the Unitas training series in Information Systems, 4 April 1997
- Office, Audit. (1989). Report of the Auditor General 1988/1989. Tasmanian State Service.
- Office, Audit. (1991). Report of the Auditor General on the Audit of Public Account for the year ended 30 June 1991. Tasmanian State Service.
- Office, Audit. (1992). Report of the Auditor General on the Audit of Government Departments and Public Bodies for the year ended 30 June 1992. Tasmanian State Service.
- Ollie, T. J Hoggstein et al (1991). *Information Systems Methodology: A framework for understanding* 2nd edition. Wokingham, England, IFIP/ Addison Wesley
- Oracle (20/10/1996). Oracle Services - Change Management. <http://www.oracle.com/consulting/html/chgmt.htm>.
- Orlikowski, W. J. (1988). *Information Technology in the Modern Organisation*. New York University. (PhD dissertation)
- Orlikowski, W. (1996) "Improvising Organisational Transformation Over Time: A Situated Change Perspective". *Information Systems Research* 7(1). pp 63-92.
- Orlikowski, W. J. and J. J. Baroudi (1991). "Studying Information Systems in Organisations: Research Approaches and Assumptions." *Information Systems Research* 2(1): 1-28.
- Orlikowski, W. J. and D. C. Gash (1994). "Technological Frames: Making Sense of Information Technology in Organisations." *ACM Transactions on Information Systems* 12(2): 174-207.
- Orlikowski, W. and D. Hofman (1997). "Improvising Organisational Transformation over time: A situated change perspective" *Information Systems Research* . 7 (1) 63-92.
- Orlikowski, W. J. and D. Robey (1991). "Information Technology and the Structuring of Organizations." *Information Systems Research* 2(2): 143-169.
- Pentland, B. T. (1995). "Information Systems and Organisational Learning: The social epistemology of organisational knowledge systems." *Accounting Management and Information Technology* 5(1): 1-21.
- Pettigrew, A. M. (1985). *The Awakening Giant: Continuity and Change in Imperial Chemical Industries*. Oxford, Basil Blackwell.
- Pettigrew, A. M. (1995). Longitudinal Field Research on Change: Theory and Practice. *Longitudinal Field Research Methods: Studying Processes of Organisational Change* Eds. G. P. Huber and A. H. Van de Ven. London, Sage. 91-125.
- Pfeffer, J (1981) *Power in Organizations* , Marchfield Mass, Pitman

- Pfeffer, J. and H. Leblebici (1977). "Information Technology and Organisational Structure." *Pacific Sociological Review* 1977(April): 241-259.
- Pfeffer, J. and G. Salancik (1990). The design and management of externally controlled organisations. *Organization theory* Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 146-178.
- Pichault, F. (1995). "The management of politics in technically related organisational change." *Organization Studies* 16(3): 449-476.
- Pinto, J. K. and J. E. Prescott (1990). "Planning and tactical factors in the project implementation process." *Journal of management studies* 27(3): 305-327.
- Poole, M. S. and A. H. Van de Ven (1989). "Using Paradox to Build Management and Organization Theories." *Academy of Management Review* 14(4): 562-578.
- Powers, M., P.H. Cheney and G.B. Crow (1990) *Structured systems development: analysis, design and implementation*. Boyd and Fraser, Boston.
- Prasad, P. (1993). "Symbolic processes in the implementation of technological change: a symbolic interactionist study of work computerisation." *Academy of management journal* 36(6): 1400-1429.
- Purvis, R. and Samburthy, V. (1997) "An examination of designer and user perceptions of JAD and the traditional IS design methodology." *Information and Management* 32 (1997) 123-135.
- Quinn, J. B. (1989). "Strategic Change: Logical Incrementalism." *Sloan Management Review* 1989(summer): 45-60.
- Rama Devi, V.S. (1995). "The Importance of Legislative Drafters". *The Loophole*. (Newsletter of the Commonwealth Association of Legislative Counsel) July 1995. pp 8-11
- Reed, M. (1991). "Review Article: Organizations and Rationality: the odd couple." *Journal of Management Studies* 28(5): 599-567.
- Robbins, S. P. and N. Barnwell (1994). *Organisation Theory in Australia*. Sydney, Prentice Hall.
- Robey, D. (1981). "Computer Information Systems and Organisational Structure." *Communications of the ACM* 24(10): 679-686.
- Robey, D. and A. Azevedo (1994). "Cultural Analysis of the Organizational Cosequences of Information Technology." *Accounting, Management and Information Technology* 4(1): 23-37.
- Robey, D. and A. Rodriguez-Diaz (1989). "The organizational and cultural context of systems implementation: case studies from Latin America." *Information and management* 17: 229-239.
- Romm, C. T., N. Plisken, et al. (1995). "The Relevance of Organizational Culture to the Implementation of Human Resource Information Systems." *Asia Pacific Journal of Human Resources* 1995(Winter): 63-80.
- Romm, T., N. Pliskin, et al. (1991). "Identifying organizational culture clash in MIS implementation." *Information and Management* 21: 99-109.
- Rusli, A. and P. Marshall (1995). *Using ethnographic inquiry methodology in IS research: an ethnographic experience in an Indonesian Organisation*. 6th Australasian conference on information systems, Perth, School of information systems, Curtin University.
- Sabherwal, R. and D. Robey (1995). An empirical taxonomy of implementation processes based on sequences of events in information systems development. *Longitudinal Field Research Methods: Studying Processes of Organisational Change* Ed. G. P. Huber and A. H. Van de Ven. London, Sage. 228-266.
- Sabherwal, R. and Elam, E. (1996) "Overcoming the problems in information systems development by building and sustaining commitment" *Accounting, Management and Information Technology*. 5 (3/4). pp 283-309
- Salmon, P. and D. Proud (1993). *Human Resource Management Systems Course notes: Managing Human Resource Information Systems, a collection of published papers and articles*. Melbourne Australia, Monash University, David Syme Faculty of Business, School of Management. 152.
- Sankar, Y. (1988). "Organizational Culture and New Technologies." *Journal of Systems Management* 1988(April): 10-17.

- Sauer, C. (1993). *Why Information Systems Fail: A Case study approach*. Henley-on-Thames, UK, Alfred Waller Ltd.
- Scott Morton, M. (ed) (1991). *The Corporation of the Nineties* Oxford, Oxford University Press.
- Scott Poole, M. and G. DeSanctis (1990). Understanding the use of Group Decision Support Systems: The theory of Adaptive structuration. *Organizations and Communication Technology* Eds. J. Fulk and C. Steinfield. Newbury Park, Sage.
- Select Databases (1997) <http://scaleplus.law.gov.au/cgi.bin/dbselect.pl/CurrentPreferences> (accessed 3/7/1997)
- Shanks, G., B. Hodgson, et al. (1996). *The use of structuration theory in information systems research*. ACIS'96, Hobart, Tasmania, ACIS.
- Shelly, G., T. Cashman, et al. (1991). *Systems analysis and design*. Boston, Boyd and Fraser.
- Shelly, G., T. Cashman, et al. (1995). *Systems analysis and design*. (2nd edition) Boston, Boyd and Fraser.
- Siddiqi, J and Shekaran, MC (1996). Requirements Engineering: The emerging Wisdom *IEEE Software*. March. 15-19.
- Smith, Andy (1997). *Human Computer Factors: A study of users and information systems*. London. McGraw-Hill
- Smyrk, J. (1997). Business Case Development. Sigma Management Services.
- Smyrk, J. (1997). Project Management. Sigma Management Science Pty Ltd.
- Snellen, I. T. M. and J. T. Schokker (1992). *Legal Application Systems in Public Administration -some specific building requirements*. European Group of Public Administration: Study Group on Informatization in Public Administration, Pisa, Italy,
- Steinfeld, C. W. and J. Fulk (1990). Theorizing about Information Technology in Organisations: The Theory Imperative. *Organisations and Communications Technology* Eds. J. Fulk and C. Steinfield. Newbury Park, London, New Delhi, Sage Publications.
- Strauss, A. (1978). *Negotiations: Varieties, Contexts, Processes and Social Order*. San Francisco, Jossey Bass.
- Strauss, A. and J. Corbin (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, Calif, Sage.
- Strauss, A. and J. Corbin. (1994) "Grounded theory methodology: an overview". N.K. Denzin and Y.S. Lincoln *Handbook of Qualitative Research* London. Sage. pp 273-285.
- Taylor, D. A. (1992). *Object-oriented information systems: planning and design*. Wiley.
- Taylor, F. (1912). Scientific Management. *Organization theory* (1990) Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 203-222.
- TheHunterGroup (13/1/97). Change Management Services. <http://www.hunter.group.com/thg/chgmt.htm>.
- Thompson, R. (1993). *Third wave project management: A Handbook for managing the complex information systems for the 1990s*. Engelwood Cliffs, Yourdon Press Computing Series.
- Tibosch, M. J. and M. S. Heng (1994). "Information Systems and Organizational Culture: A viewpoint." *Malaysian Journal of Management Science* 3(2): 17-33.
- Trist, E. L. (1981). The Socio-technical perspective: The Evolution of Sociotechnical Systems as a conceptual framework and as an action research program. *Perspectives on Organization Design and Behavior* Ed. A. Van Den Ven and J. WF. USA, John Wiley and Sons.
- Truex, D. (1993). Information Systems Development in the Emergent Organisation. University of Binghamton. (PhD dissertation)
- Turner, J. A. (1987). Understanding the elements of system design. *Critical issues in information systems research* Ed. J. Boland and R. Hirschheim. John Wiley and Sons. 79-96.

- Turner, JR (1993) *The Handbook of Project Based Management*. London, McGraw-Hill.
- Van De Ven, A. H. and W. F. Joyce (1981). *Perspectives on Organisation Design and Behavior*. USA, Canada, John Wiley and Sons.
- van den Berg, A. (1980). "Critical Theory: Is there still hope?" *American Journal of Sociology* 86(3): 449-478.
- Van Maanen, J. (1991). Playing Back the Tape: Early Days in the Field. *Experiencing Fieldwork: An Inside View of Qualitative Research* Eds. W. B. Shaffir and R. A. Stebbins. Newbury Park, Sage.
- van Slooten, K. and B. Schoonhoven (1994) "Towards contingent information systems development approaches." *Proceedings 4th International Conference on Information Systems*. Bled Slovenia 20-22 September 1994. 242-254.
- Vidgen, R. and T. McMaster (1995). *Black boxes, non-human stakeholders and the translation of IT through mediation*. IFIP, Cambridge, UK
- Villiers, A. (1955) *The Set of the Sails* London. Hodder and Stoughton Ltd
- Vitalari, N. P. (1985). The Need for Longitudinal Designs in the Study of Computing Environments. *Research Methods in Information Systems* Eds. E. Mumford. North Holland, Elsevier.
- Voermans, W. and E. Verharen (1993). LEDA: A semi-intelligent legislative drafting-support system. *Legal knowledge based systems* Ed. J. S. Svensson, J. G. J. Wassink and B. v. Buggenhout. Koninlijke vermande. 81-94.
- Walsham, G. (1992). "Management Science and Organisational Change: A Framework for Analysis." *OMEGA International Journal of Management Science* 20(1): 1-9.
- Walsham, G. (1993). *Interpreting Information Systems in Organisations*. Chichester, John Wiley.
- Walsham, G. (1993). "Reading the organisation: metaphores and information management." *Journal of Information Systems* 3(1): 33-46.
- Walsham, G. (1995). "The Emergence of Interpretivism in IS research." *Information Systems Research* 6(4): 376-394.
- Walsham, G. (1995). "Interpretive case studies in IS research: nature and method." *European Journal of Information Systems* 1995(4): 74-81.
- Walsham, G. and H. Chun-Kwong (1991). "Structuration Theory and Information Systems Research." *Journal of Applied Systems Analysis* 17: 77-85.
- Walsham, G. and C. K. Han (1993). "Information Systems Strategy formation and implementation: the case of a central government agency." *Accounting Management and Information Technology* 3(3): 191-209.
- Walsham, G. and S. Sunday (1996). GIS for District-Level Administration in India: Problems and Opportunities. Lancaster University Management School. (working paper)
- Walsham, G. and T. Waema (1994). "Information Systems Strategy and Implementation: A Case Study of a Building Society." *ACM Transactions on Information Systems* 12(2): 150-173.
- Ward, J. and P. Griffiths (1996). *Strategic Planning for Information Systems*. Chichester, UK, John Wiley and Sons.
- Ward, J., P. Griffiths, et al. (1990). *Strategic Planning for Information Systems*. Chichester, UK, John Wiley and Sons.
- Ward, P. (1981). "Computerisation of Legal Material in Australia." *JLIS* 1(2): 162-174.
- Warne, L. (1995). *Organisational power and information systems development: findings from a case study of a large public sector project*. 6th Australasian conference on information systems, Perth, School of information systems, Curtin University.
- Warner Burke, W. (1994). Critical Elements of Organizational Culture Change. *The Change Management Handbook: A road map to corporate transformation* Ed. L. A. Berger, M. J. Sikora and D. R. Berger. Burr Ridge, Illinois, Irwin Professional Publishing. 285-294.

- Weaver, G. and D. Gioia (1994). "Paradigms Lost: Incommensurability and Structural Inquiry." *Organization Studies* 15(4): 565-590.
- Weber, M. (1924). Legitimate authority and bureaucracy. *Organization theory* Ed. D. S. Pugh. Harmondsworth, UK, Penguin. 3rd, ed. 3-16.
- Westrup, C. (1995). *Transforming Organizations through Systems Analysis: Deploying new techniques for Organizational Analysis in IS Development*. Information Technology and Changes in Organisational Work, Cambridge, Chapman and Hall.
- Wetherbe, J. C. (1991). "Executive Information Requirements: Getting it right." *MISQ* 15(1): 51-85.
- Whiteley, A. (1995). *Managing change: a core values approach*. Melbourne, Macmillan.
- Willmott, H. (1993). "Breaking the paradigm mentality." *Organization Studies* 14(5): 681-719.
- Wolfe, R. A. (1994). "Organizational Innovation: Review, Critique and Suggested Research Directions." *Journal of Management Studies* 31(3): 405-431.
- Wood-Harper, A.T. Antill, L. and Avison, D.E. (1985) *Information Systems Definition: The Multiview Approach* Oxford, Blackwell Scientific.
- WorthinghamBrighton.Press (13/1/97). The Change Management Toolkit.
<http://www.utsi.com/wbs/reengineering/preface.html>.
- Wynekoop J. and N. Russo (1997). "Studying system development methodologies: an examination os research methods." *Information Systems Journal* 7 47-65.
- Yates, J. and R. I. Benjamin (1991). The Past and Present as a Window on the Future. *The Corporation of the Nineties* Ed. T. J. Allen and M. S. Scott Morton. Oxford, Oxford University Press. 61-92.
- Yetton, P., J. Craig, et al. (1995). *Fit, simplicity and risk: multiple paths to strategic IT change*. ACIS (submitted to), Amsterdam.
- Yetton, P. W. and K. D. Johnston (1993). *Information Technology and the Dynamics of Strategic Fit*. ACIS, Brisbane,
- Yin, R. K. (1989). *Case Study Research: Design and Methods*. Newbury Park, Sage Publications.
- Yin, R. K. (1989). Research Design Issues in Using the Case Study Method to Study Management Information Systems. *The Information Systems Research Challenge: Qualitative Research Methods* Eds. J. I. J. Cash and P. R. Lawrence. Boston, Harvard Business School Press.
- Younis, T. and I. Davidson (1990). The Study of Implementation. *Implementation in Public Policy* Ed. T. Younis. Aldershot USA, Dartmouth. 3-14.
- Yourdan, (1989) *Modern Structured Analysis* Engelwood Cliffs, NJ USA. Prentice Hall.